



André Filipe dos Reis Rodrigues

Licenciatura em Ciência e Engenharia Informática

Aplicação Móvel de Processamento de Imagem para Análise Quantitativa de Biossensores

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador: Nuno Manuel Robalo Correia, Professor Catedrático,
Universidade Nova de Lisboa

Co-orientadora: Elvira Maria Correia Fortunato, Professora Catedrática,
Universidade Nova de Lisboa

Júri

Presidente: Doutor Pedro Manuel Corrêa Calvente de Barahona
Vogal: Doutor Manuel João Caneira Monteiro da Fonseca



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Dezembro, 2018

Aplicação Móvel de Processamento de Imagem para Análise Quantitativa de Biossensores

Copyright © André Filipe dos Reis Rodrigues, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

AGRADECIMENTOS

Em primeiro lugar, gostaria de agradecer aos meus orientadores, Professor Doutor Nuno Correia e Professora Doutora Elvira Fortunato, por todo o apoio, opiniões, críticas, sugestões e dedicação durante a realização desta dissertação.

Gostaria também de agradecer à aluna de doutoramento do *CENIMAT* Ana Marques, ao Mestre em Engenharia Biomédica João Ferrão e ao aluno de Engenharia Biomédica Tomás Pinheiro por todo o apoio, dedicação e fornecimento de materiais importantes relacionados com a temática de biossensores em que esta dissertação se enquadra.

Agradeço também ao colega de curso Gonçalo Moncada por todo o apoio prestado na área de *Machine Learning* e aos colegas Henrique Rodrigues, Bárbara Silveira e Raquel Macedo por todo o apoio e companheirismo durante este trabalho.

Deixo o meu agradecimento a todos os que participaram nos testes de usabilidade da aplicação e que ajudaram a avaliar o que foi desenvolvido.

Finalmente, agradeço à minha mãe e à minha família por todo o apoio e motivação que me deram durante este período.

Do or do not, there is no try

RESUMO

Sendo a prevenção e detecção de doenças uma parte fundamental da manutenção da saúde das populações por todo o planeta, é natural que seja colocado um grande ênfase nas tecnologias que permitem o seu diagnóstico. No entanto, verifica-se uma situação de carência dessas tecnologias, e com isso surge a necessidade de desenvolver meios de diagnóstico inovadores através de diversas tecnologias como a emergente *Lab-on-Paper*, uma tecnologia baseada em reações químicas sobre papel.

Com este material foi produzido um biossensor colorimétrico de leitura da concentração de glucose no sangue, indicada pela cor assumida pelo papel, de modo a melhorar o diagnóstico e a prevenção de uma das doenças mais incidentes em todo o mundo, a *Diabetes Mellitus*.

Assim, em parceria com o CENIMAT (Centro de Investigação de Materiais), foi desenvolvida uma aplicação *Android* com a capacidade de realizar a análise colorimétrica deste biossensor. A análise é realizada através de diferentes técnicas de Processamento de Imagem em conjunto com dois modelos de *Machine Learning*: um para medir diabetes em jejum e outro para medir 2 horas após as refeições, através da cor do sensor capturada pela câmara do dispositivo móvel.

Pelos testes realizados, esta aplicação é capaz de detetar corretamente o biossensor a elevada velocidade, seguido por um passo de calibração de cores com base em zonas de referência. No entanto, de modo a melhorar os resultados é necessário adicionar mais zonas de referência com diferentes cores, assim como melhorar os resultados do biossensor de modo a obter um melhor diagnóstico.

Além de testes às funcionalidades, foi também realizado um estudo com possíveis utilizadores com base num questionário de usabilidade. Os resultados deste estudo demonstraram que os utilizadores ficaram bastante satisfeitos com o que a aplicação disponibiliza, tendo alguns dado sugestões para informação adicional da análise dos diabetes.

Palavras-chave: Aplicação Móvel, Processamento de Imagem, Análise Colorimétrica, Classificação, Biossensor, Papel, Diabetes

ABSTRACT

Since disease prevention and detection are fundamental parts in maintaining worldwide population health, it is only natural for there to be great emphasis put into developing diagnostic technologies. There is, therefore, a need to create innovative means of diagnosis through many different technologies like the emerging Lab-on-Paper, which is based in chemical reactions on paper as the name indicates.

With this material, a colorimetric biosensor to measure glucose concentration in blood samples through color was produced with the purpose of improving the diagnosis of one of the most active diseases globally, Diabetes Mellitus.

With this in mind, and together with *CENIMAT* (Centro de Investigação de Materiais), an *Android* application was developed in order to perform colorimetric analysis of said biosensor through Image Processing techniques and two *Machine Learning* models for performing a colorimetric analysis: one for a fasting state analysis and another for 2 hours after meals. This is in order to facilitate the process of reading diabetes values, through the sensor's color captured by the mobile device's camera.

Results from testing determined that the application is capable of correctly detecting the biosensor, as well as perform a highly efficient colorimetric analysis, in combination with an image color calibration step based on color reference zones located on the biosensor. The testing also found the need to increase the number of color reference zones, as well as the need to improve the colorimetric output of the biosensor, in order to better distinguish different glucose concentrations and improve the application's diagnostic.

Besides functionality testing, a study related to the usability of the user interface based on user questionnaire was performed, in order to better satisfy the needs of users for this system. Results showed that, overall users were quite satisfied with how they interacted with the application, while some gave a few suggestions of what extra information should the application give, related to the analysis result.

Keywords: Mobile Application, Image Processing, Colorimetric Analysis, Classification, Biosensor, Paper, Diabetes

ÍNDICE

Lista de Figuras	xix
1 Introdução	1
1.1 Motivação	1
1.2 Contexto	2
1.3 Principais Objetivos e Contribuições	3
1.4 Organização do Documento	4
2 Enquadramento	5
2.1 Processamento de Imagem	5
2.1.1 Filtros de convolução	6
2.1.2 Cor	6
2.1.3 Forma	10
2.1.4 Classificação	13
2.2 Detecção e Quantificação Colorimétrica	17
2.2.1 Quantificação a partir de <i>smartphones</i>	17
2.3 Usabilidade	18
2.3.1 Avaliação Heurística	18
2.3.2 Questionários de Usabilidade	18
2.3.3 <i>Design</i> Gráfico	19
3 Trabalho Relacionado	23
3.1 Análise de Suor	23
3.1.1 Captura da Imagem do Biossensor	24
3.1.2 Quantificação Colorimétrica da Análise do Suor	25
3.1.3 Resultados	25
3.2 Análise de Urina	26
3.2.1 Calibração	26
3.2.2 Análise Colorimétrica do Biossensor	26
3.2.3 Resultados	27
3.3 Detecção Colorimétrica com <i>Machine Learning</i>	28
3.3.1 <i>Design</i> Experimental	29
3.3.2 Captura de Imagens	29

3.3.3	Pré-Processamento	29
3.3.4	Extração de Características	30
3.4	Avaliação Crítica	33
4	Desenho da Aplicação	35
4.1	Funcionalidades e Cenários	35
4.2	Tecnologias e Arquitetura da Aplicação	37
5	Implementação	39
5.1	Protótipo	39
5.1.1	Deteção do Biossensor	40
5.1.2	Análise Colorimétrica	42
5.1.3	Interface Gráfica	43
5.2	Aplicação Final	44
5.2.1	Deteção do Biossensor	44
5.2.2	Análise Colorimétrica	48
5.2.3	Interface Gráfica	54
6	Testes e Avaliação	55
6.1	Protótipo	55
6.1.1	Deteção do Biossensor	55
6.1.2	Análise Colorimétrica	56
6.2	Aplicação Final	56
6.2.1	Deteção do Biossensor	56
6.2.2	Calibração de Cor	57
6.2.3	Análise Colorimétrica	57
6.2.4	Testes de Integração das Componentes	60
6.2.5	Teste de Utilizadores	61
7	Conclusões e Trabalho Futuro	63
	Bibliografia	65
A	<i>Storyboard da Aplicação</i>	69
B	<i>Interface Gráfica da Mellitus</i>	71
B.1	Ecrãs Principais	71
B.2	Ecrãs do <i>Tutorial</i>	73
B.3	Ecrãs de Depuração	74
C	Resultados dos Testes de Deteção do Biossensor e Calibração de Cor	75
C.1	Deteção do biossensor	76
C.1.1	Luz Solar	76

C.1.2	Luz Fluorescente	84
C.1.3	Luz Incandescente	87
C.2	Calibração da cor	96
C.2.1	Luz Solar	96
C.2.2	Luz Fluorescente	97
C.2.3	Luz Incandescente	97
D	Resultados de <i>Cross-Validation</i>	99
D.1	<i>Cross-Validation</i> com RGB	100
D.2	<i>Cross-Validation</i> com YUV	100
D.3	<i>Cross-Validation</i> com HLS	101
D.4	<i>Cross-Validation</i> com HSV	101
D.5	<i>Cross-Validation</i> com XYZ	102
D.6	<i>Cross-Validation</i> com o rácio de R com G	102
E	Regiões de Decisão dos Modelos Desenvolvidos	103
E.1	Modelo de cor RGB	104
E.1.1	Concentrações de glucose	104
E.1.2	Estado de Jejum	105
E.1.3	Duas horas após a refeição	107
E.2	Modelo de cor YUV	108
E.2.1	Concentrações de glucose	108
E.2.2	Estado de Jejum	110
E.2.3	Duas horas após a refeição	111
E.3	Modelo de cor HSV	113
E.3.1	Concentrações de glucose	113
E.3.2	Estado de Jejum	114
E.3.3	Duas horas após a refeição	116
E.4	Modelo de cor HLS	117
E.4.1	Concentrações de glucose	117
E.4.2	Estado de Jejum	119
E.4.3	Duas horas após a refeição	120
E.5	Modelo de cor XYZ	122
E.5.1	Concentrações de glucose	122
E.5.2	Estado de Jejum	123
E.5.3	Duas horas após a refeição	125
E.6	Modelo a partir do rácio de R com G	126
E.6.1	Concentrações de glucose	126
E.6.2	Estado de Jejum	128
E.6.3	Duas horas após a refeição	129
F	Questionário de Usabilidade	131

F.1	Qual a sua idade?	131
F.2	É Diabético/a?	132
F.3	Acho que a deteção do biossensor é fácil	132
F.4	Inicialmente tive dificuldade em descobrir como detetar o sensor	133
F.5	A informação da análise é útil	134
F.6	A aplicação é mais fácil de se usar que uma máquina de diabetes normal	134
F.7	Os resultados da análise da aplicação são mais úteis que os de uma máquina normal	135
F.8	Considero que a aplicação facilita a análise aos diabetes	136

LISTA DE FIGURAS

2.1	Diagrama cromático <i>CIE XYZ</i> . A fronteira do diagrama representa a correspondência da cromaticidade com o comprimento de onda de luz monoespectral em nanómetros. De notar que este corte de $x + y + z = 1$ não representa todas as cores existentes. [12].	7
2.2	Paleta de cores do modelo RGB [24].	8
2.3	Paleta de cores do modelo CMY [24].	8
2.4	Uma imagem e o seu respetivo histograma global, assim como algumas medidas com base nos <i>pixels</i> . [24].	9
2.5	a) Um ponto (x_0, y_0) no plano da imagem. b) As diferentes linhas parametrizadas por (ρ_i, θ_i) para o ponto. c) Curva do plano acumulador (ρ, θ) para o ponto da figura a) [7].	11
2.6	Imagem original e resultado da aplicação do algoritmo <i>Canny</i> [7].	12
2.7	Visualização gráfica de uma classificação realizada com <i>SVM</i> de duas classes de dados distintas em cada lado do <i>hiperplano</i> [2].	15
2.8	Exemplo de uso de <i>Material Design</i> em <i>Rally</i> , uma aplicação de gestão de finanças pessoais [9].	21
3.1	Biossensor de captura de suor e as suas respetivas zonas de cor [13].	24
3.2	Processo de captura de valores periódicos pelo dispositivo móvel [13].	24
3.3	Curvas de calibração de cada concentração e pH a calcular. Cada linha representa o valor do canal de cor no eixo das ordenadas e as abcissas o valor a calcular da substância [13].	25
3.4	Representação de alguns valores calculados e da curva de calibração no gráfico cromático 2D [26].	28
3.5	Comparação dos resultados medidos pelos dois tipos diferentes de <i>smartphones</i> com o seu respetivo erro [26].	28
3.6	a) Influência de iluminações diferentes nas cores numa tira de cor de pH 12.0. b) Diferentes tiras para valores diferentes de pH capturadas em diferentes formatos de imagem com o dispositivo de impressão 3D [15].	30
3.7	Resultados de precisão da deteção automática com e sem isolamento de luz ambiente [15].	31
3.8	Representação dos resultados dos testes do classificador [15].	32

4.2	Arquitetura das componentes de <i>Software</i> da aplicação móvel.	37
4.3	Diagrama de componentes da aplicação.	38
5.1	Biossensor usado na aplicação de protótipo.	40
5.2	Aplicação desenvolvida e os resultados de duas experiências.	41
5.3	Poços preenchidos com cor representativa de concentração de 10 mM (mili-molares) de glucose.	42
5.4	Rácio entre o canal Vermelho e o canal Verde da média dos momentos de cor registados para cada concentração de glucose.	42
5.5	Novo sensor usado para a aplicação final.	44
5.6	Imagem normal e o resultado do filtro de laplace, apresentando variância de 22.8.	45
5.7	Imagem desfocada devido ao movimento da câmara e o resultado do filtro de laplace, apresentando variância de 13.19.	45
5.8	Imagem desfocada e o resultado do filtro de laplace, apresentando variância de 11.53.	46
5.9	Comparação da imagem da zona do poço do biossensor em cinzento, de notar as cores de calibração na implementação.	47
5.10	Comparação da imagem da zona do poço do biossensor binarizada, após a conversão para cinzento original e da implementada, sem aumento do brilho.	47
5.11	Comparação da imagem da zona do poço do biossensor em cinzento após a conversão para cinzento original e da implementada a operação de aumento do brilho.	47
5.12	Comparação da imagem da zona do poço do biossensor binarizada, após a conversão para cinzento original e da implementada, com aumento do brilho.	48
5.13	Ecrã de deteção do sensor.	49
5.14	Diferença da calibração de cor antes e depois de aplicada a transformação em cada pixel na região do poço, tirado com <i>Samsung Galaxy S8</i>	50
5.15	Coleção de diferentes poços para a concentração de 5 mM.	52
5.16	Projeção <i>PCA</i> do conjunto de dados com o modelo <i>RGB</i> do biossensor.	52
5.17	Projeção <i>PCA</i> do conjunto de dados com o modelo <i>RGB</i> do biossensor agrupados com base na tabela 5.3, 2 horas após refeição.	53
5.18	Projeção <i>PCA</i> do conjunto de dados com o modelo <i>RGB</i> do biossensor agrupados com base na tabela 5.3, em estado de jejum.	53
6.1	Regiões de decisão do modelo para utilizador em estado de jejum.	60
6.2	Regiões de decisão do modelo para utilizador 2 horas após refeição.	61
A.1	70
B.1	Interfaces gráficas principais da aplicação.	72
B.2	Interfaces gráficas do <i>tutorial</i>	73

B.3 Interfaces gráficas adicionais de depuração.	74
C.1 Biossensor vazio a luz natural.	76
C.2 Biossensor com 1.5 mM a luz natural.	76
C.3 Biossensor com 5 mM a luz natural.	76
C.4 Biossensor com 20 mM a luz natural.	76
C.5 Operações de brilho que não detetaram o sensor vazio.	77
C.6 Operações de brilho que detetaram o sensor vazio.	77
C.7 Operações de brilho que detetaram o sensor de 1.5 mM.	77
C.8 Operações de brilho que não detetaram o sensor de 5 mM.	77
C.9 Operações de brilho que detetaram o sensor de 5 mM.	78
C.10 Operações de brilho que não detetaram o sensor de 20 mM.	78
C.11 Operações de brilho que detetaram o sensor de 20 mM.	78
C.12 Operações de binarização que não detetaram o sensor vazio.	79
C.13 Operações de binarização que detetaram o sensor vazio.	79
C.14 Operações de binarização que detetaram o sensor com 1.5 mM.	79
C.15 Operações de binarização que não detetaram o sensor com 5 mM.	79
C.16 Operações de binarização que não detetaram o sensor com 5 mM.	80
C.17 Operações de binarização que não detetaram o sensor com 20 mM.	80
C.18 Operações de binarização que detetaram o sensor com 20 mM.	80
C.19 Círculos totais obtidos com $r = 1$ no sensor vazio.	81
C.20 Círculos totais obtidos no sensor vazio.	81
C.21 Círculos totais obtidos no sensor com 1.5 mM.	81
C.22 Círculos totais obtidos com $r = 1$ no sensor de 5 mM.	82
C.23 Círculos totais obtidos no sensor de 5 mM.	82
C.24 Círculos totais obtidos com $r = 1$ no sensor de 20 mM.	82
C.25 Círculos totais obtidos no sensor de 20 mM.	82
C.26 Círculos finais obtidos com $r = 1$	83
C.27 Círculos finais obtidos.	83
C.29 Círculos finais obtidos com $r = 1$ no sensor de 5 mM.	83
C.30 Círculos finais obtidos no sensor de 5 mM.	84
C.31 Círculos finais obtidos com $r = 1$ no sensor de 20 mM.	84
C.32 Círculos finais obtidos no sensor de 20 mM.	84
C.33 Biossensor vazio a luz fluorescente.	84
C.34 Operações de brilho que não detetaram o sensor.	85
C.35 Operações de brilho que detetaram o sensor.	85
C.36 Operações de binarização que não detetaram o sensor.	85
C.37 Operações de binarização que detetaram o sensor.	85
C.38 Círculos totais obtidos com $r = 1$	86
C.39 Círculos totais obtidos.	86
C.40 Círculos finais obtidos com $r = 1$	86

C.41 Círculos finais obtidos.	86
C.42 Biossensor vazio a luz incandescente.	87
C.43 Biossensor com 1.5 mM a luz natural.	87
C.44 Biossensor com 5 mM a luz natural.	87
C.45 Biossensor com 20 mM a luz natural.	87
C.46 Operações de brilho com $r = 1$ que não detetou o biossensor.	88
C.47 Operações de brilho que detetaram o sensor.	88
C.48 Operações de brilho que não detetaram o sensor de 1.5 mM.	88
C.49 Operações de brilho que detetaram o sensor de 1.5 mM.	88
C.50 Operações de brilho que não detetaram o sensor de 5 mM.	89
C.51 Operações de brilho que detetaram o sensor de 5 mM.	89
C.52 Operações de brilho que não detetaram o sensor de 20 mM.	89
C.53 Operações de brilho que detetaram o sensor de 20 mM.	89
C.54 Operações de binarização que não detetaram o sensor.	90
C.55 Operações de binarização que detetaram o sensor.	90
C.56 Operações de binarização que não detetaram o sensor de 1.5 mM.	90
C.57 Operações de binarização que detetaram o sensor de 1.5 mM.	90
C.58 Operações de binarização que não detetaram o sensor de 5 mM.	91
C.59 Operações de binarização que detetaram o sensor de 5 mM.	91
C.60 Operações de binarização que não detetaram o sensor de 20 mM.	91
C.61 Operações de binarização que detetaram o sensor de 20 mM.	91
C.62 Círculos totais obtidos com $r = 1$ no sensor vazio.	92
C.63 Círculos totais obtidos no sensor vazio.	92
C.64 Círculos totais obtidos no sensor de 1.5 mM.	92
C.65 Círculos totais obtidos no sensor de 1.5 mM.	93
C.66 Círculos totais obtidos no sensor de 5 mM.	93
C.67 Círculos totais obtidos no sensor de 5 mM.	93
C.68 Círculos totais obtidos no sensor de 20 mM.	93
C.69 Círculos totais obtidos no sensor de 20 mM.	94
C.70 Círculos finais obtidos com $r = 1$	94
C.71 Círculos finais obtidos.	94
C.72 Círculos finais obtidos no sensor de 1.5 mM.	95
C.73 Círculos finais obtidos no sensor de 1.5 mM.	95
C.74 Círculos finais obtidos no sensor de 5 mM.	95
C.75 Círculos finais obtidos no sensor de 5 mM.	95
C.76 Círculos finais obtidos no sensor de 20 mM.	96
C.77 Círculos finais obtidos no sensor de 20 mM.	96
C.78 Comparação das cores antes e após a operação de calibração ao sensor vazio.	96
C.79 Comparação das cores antes e após a operação de calibração ao sensor com 1.5 mM.	96

C.80 Comparação das cores antes e após a operação de calibração ao sensor com 5 mM.	97
C.81 Comparação das cores antes e após a operação de calibração ao sensor com 20 mM.	97
C.82 Comparação das cores antes e após a operação de calibração a luz fluorescente.	97
C.83 Comparação das cores antes e após a operação de calibração a luz incandescente.	97
C.84 Comparação das cores antes e após a operação de calibração ao sensor com 1.5 mM.	98
C.85 Comparação das cores antes e após a operação de calibração ao sensor com 5 mM.	98
C.86 Comparação das cores antes e após a operação de calibração ao sensor com 20 mM.	98
E.1 <i>Support Vector Machines</i>	104
E.2 <i>K-Nearest Neighbors</i>	104
E.3 <i>Random Trees</i>	105
E.4 <i>Support Vector Machines</i>	105
E.5 <i>K-Nearest Neighbors</i>	106
E.6 <i>Random Trees</i>	106
E.7 <i>Support Vector Machines</i>	107
E.8 <i>K-Nearest Neighbors</i>	107
E.9 <i>Random Trees</i>	108
E.10 <i>Support Vector Machines</i>	108
E.11 <i>K-Nearest Neighbors</i>	109
E.12 <i>Random Trees</i>	109
E.13 <i>Support Vector Machines</i>	110
E.14 <i>K-Nearest Neighbors</i>	110
E.15 <i>Random Trees</i>	111
E.16 <i>Support Vector Machines</i>	111
E.17 <i>K-Nearest Neighbors</i>	112
E.18 <i>Random Trees</i>	112
E.19 <i>Support Vector Machines</i>	113
E.20 <i>K-Nearest Neighbors</i>	113
E.21 <i>Random Trees</i>	114
E.22 <i>Support Vector Machines</i>	114
E.23 <i>K-Nearest Neighbors</i>	115
E.24 <i>Random Trees</i>	115
E.25 <i>Support Vector Machines</i>	116
E.26 <i>K-Nearest Neighbors</i>	116
E.27 <i>Random Trees</i>	117
E.28 <i>Support Vector Machines</i>	117

E.29	<i>K-Nearest Neighbors</i>	118
E.30	<i>Random Trees</i>	118
E.31	<i>Support Vector Machines</i>	119
E.32	<i>K-Nearest Neighbors</i>	119
E.33	<i>Random Trees</i>	120
E.34	<i>Support Vector Machines</i>	120
E.35	<i>K-Nearest Neighbors</i>	121
E.36	<i>Random Trees</i>	121
E.37	<i>Support Vector Machines</i>	122
E.38	<i>K-Nearest Neighbors</i>	122
E.39	<i>Random Trees</i>	123
E.40	<i>Support Vector Machines</i>	123
E.41	<i>K-Nearest Neighbors</i>	124
E.42	<i>Random Trees</i>	124
E.43	<i>Support Vector Machines</i>	125
E.44	<i>K-Nearest Neighbors</i>	125
E.45	<i>Random Trees</i>	126
E.46	<i>Support Vector Machines</i>	126
E.47	<i>K-Nearest Neighbors</i>	127
E.48	<i>Random Trees</i>	127
E.49	<i>Support Vector Machines</i>	128
E.50	<i>K-Nearest Neighbors</i>	128
E.51	<i>Random Trees</i>	129
E.52	<i>Support Vector Machines</i>	129
E.53	<i>K-Nearest Neighbors</i>	130
E.54	<i>Random Trees</i>	130
F.1		132
F.2		132
F.3		133
F.4		133
F.5		134
F.6		135
F.7		135
F.8		136

INTRODUÇÃO

Neste capítulo é descrita a motivação e o contexto desta dissertação, assim como as principais contribuições para o projeto descrito, definindo o respetivo problema de forma pragmática. Por fim é também apresentada a estrutura deste documento, descrevendo o conteúdo de cada capítulo em conjunto com os apêndices.

1.1 Motivação

Na área da saúde, um dos aspetos mais importantes na prevenção e tratamento de doenças consiste na facilidade de acesso a meios de diagnóstico simples com o objetivo de as detetar o mais rapidamente possível. Os meios de diagnóstico são cruciais para a identificação e causas de doenças tanto a nível individual como populacional, potencialmente seguidos por um procedimento de tratamento adequado a cada doença e a posterior monitorização dos seus efeitos.

No entanto, embora estes meios de diagnóstico se encontrem geralmente disponíveis nos países considerados desenvolvidos, o mesmo não sucede em grande parte dos países em vias de desenvolvimento, sendo muitas vezes considerados pouco eficientes, desatualizados ou ausentes. No caso da sua ausência, vários profissionais de saúde apenas realizam diagnósticos e tratamentos puramente com base nos sintomas observados [10, 27].

Uma potencial solução para este problema consiste no uso do papel como material de tecnologia de diagnósticos pois este é considerado um material de baixo custo, disponível e de rápido uso. Este material possui a capacidade de detetar diversas substâncias químicas como metabolitos de urina, glucose, hormonas e vários agentes infecciosos, a partir de diversas reações químicas representadas pela cor [11].

Sendo uma das doenças mais importantes de se diagnosticar tanto nos países desenvolvidos como nos países em desenvolvimento, a Diabetes, (do nome completo *Diabetes*

Mellitus) é uma doença crónica cada vez mais frequente na sociedade, estimando-se que em 2015 existiam 415 milhões de pessoas com diabetes, tendo causado a morte de 5 milhões de pessoas no mesmo ano. Além do impacto na saúde da população, a Diabetes é também responsável por aproximadamente 12% dos gastos em saúde no mundo [18]. Como tal, devido à ausência de uma cura, o diagnóstico rápido é importante, sendo este realizado pela medição da concentração de glucose no sangue enquanto o sujeito se encontra em estado de jejum ou 2 horas após as refeições [3]. Posto isto, justifica-se a grande importância da existência de métodos de diagnóstico da Diabetes disponíveis de forma generalizada [5].

1.2 Contexto

Esta dissertação enquadra-se na continuação de outra dissertação de Mestrado da autoria de João Miguel Paitio Ferrão [6] realizada na fronteira entre Engenharia de Materiais, Engenharia de Micro e Nanotecnologias e Engenharia Biomédica. O objetivo final foi a construção de um biossensor colorimétrico não enzimático em papel para a determinação do valor de diabetes a partir da medição da concentração de glucose no sangue, usando tecnologia *Lab-on-Paper*, uma tecnologia baseada em reações químicas sobre papel. Além deste sensor ser feito com materiais de baixo custo, apresenta a grande vantagem de ao contrário dos sensores enzimáticos (máquinas de análise de diabetes convencionais), os valores a medir não serem afetados pela temperatura ambiente ou outros factores externos. No entanto, devido à dificuldade de interpretar visualmente os resultados do sensor, foram realizados testes de análise colorimétrica digital usando o *software ImageJ*¹ com base em imagens do sensor criadas por um dispositivo *scanner* e também pela câmara de um *smartphone*. Estes testes apresentaram resultados bastante positivos e precisos, destacando principalmente as imagens capturadas pelo *smartphone* por apresentarem resultados bastante semelhantes aos das imagens digitalizadas [6].

Com base na dissertação referida e em parceria com o CENIMAT (Centro de Investigação em Materiais), esta dissertação tem por objetivo construir uma aplicação móvel com o fim de facilitar e automatizar o processo de análise colorimétrica digital. Esta aplicação deve ser capaz de realizar a devida análise do sensor já desenvolvido a partir de técnicas de Processamento de Imagem, automatizando assim o processo de análise digital e mostrando no fim ao utilizador, o resultado em valores tipicamente usados por outros dispositivos de análise de diabetes.

¹<https://imagej.nih.gov/ij>

1.3 Principais Objetivos e Contribuições

Como resultado da dissertação esperam-se as seguintes contribuições principais:

Desenvolvimento de uma Aplicação móvel O objetivo principal desta dissertação consiste no desenvolvimento de uma aplicação para ser usada em dispositivos móveis, com a funcionalidade de medir o valor dos diabetes por parte dos utilizadores. Esta aplicação, em conjunto com o respetivo biossensor colorimétrico de papel, apresentam a principal vantagem de fornecer um meio de diagnóstico de custo bastante mais baixo que os meios já existentes, podendo este ser usado em centros de saúde ou integrado em sistemas informáticos médicos mais complexos. Posto isto, esta aplicação pode ser dividida nas seguintes componentes:

Interface Gráfica A interface gráfica é uma componente fundamental para qualquer aplicação, sendo essa com a qual o utilizador interage de modo a comunicar com o sistema. Para uma aplicação móvel, esta componente necessita de ser desenvolvida para uma plataforma específica, sendo que nesta dissertação a aplicação foi apenas desenvolvida na plataforma do sistema operativo *Android* ². Este sistema é conhecido por ser o sistema com maior quota de mercado do mundo [22] e além de tomar a filosofia de código aberto, assim como a maior parte dos sistemas operativos móveis possui uma loja de aplicações que permite facilmente a sua distribuição. Além disso é possível desenvolver aplicações para esse sistema em diferentes plataformas computacionais.

Algoritmo Para esta aplicação, foi desenvolvido um algoritmo que é capaz de realizar a leitura e análise colorimétrica do biossensor, tendo como base diferentes técnicas de Processamento de Imagem usando a biblioteca *OpenCV* ³, uma biblioteca de Visão por Computador e Processamento de Imagem. Além desta plataforma ser de código aberto, é compatível com diferentes plataformas móveis. Além do desenvolvimento deste algoritmo, também foi necessária a garantia do seu funcionamento correto, sendo este determinado a partir de testes com base na precisão da análise e leitura.

Estudo de Usabilidade A Usabilidade é uma propriedade importante no desenvolvimento de *software*, mais precisamente do desenvolvimento da interface gráfica. Considera-se que na maior parte dos projetos, o código da interface é aproximadamente 48% da quantidade do código fonte do *software* [16]. Devido à dimensão desta componente e de modo a desenvolver uma aplicação fácil de usar, foi efetuado um estudo, análise e avaliação da Usabilidade centrada nas tarefas que o utilizador poderá realizar na aplicação, criando assim um meio de diagnóstico simples e acessível de ser usado.

²<https://www.android.com>

³<https://opencv.org>

1.4 Organização do Documento

Este documento encontra-se dividido nos seguintes capítulos:

Capítulo 1 - Introdução É apresentado o tema desta dissertação, discutindo o seu contexto, motivação e os principais objetivos e contribuições previstas.

Capítulo 2 - Enquadramento Neste capítulo são apresentados os conceitos relevantes para o desenvolvimento da aplicação, de modo a ser possível cumprir todos os objetivos propostos.

Capítulo 3 - Trabalho Relacionado Neste capítulo são descritos vários projetos relacionados com os respetivos temas desta dissertação, comparando e discutindo-os entre eles e a aplicação desenvolvida, descrevendo a contribuição deste trabalho relativamente ao que já foi feito.

Capítulo 4 - Desenho da Aplicação É descrito todo o desenho da aplicação, apresentando as interações que o utilizador terá, seguido pelo desenho da respetiva arquitetura, discutida em conjunto com as diferentes tecnologias utilizadas.

Capítulo 5 - Implementação Neste capítulo é apresentada a implementação de todas as componentes que constituem a aplicação, descrevendo passo a passo os algoritmos usados, assim como algumas breves justificações da sua utilização.

Capítulo 6 - Testes e Avaliação Aqui são apresentados todos os testes e avaliações realizadas da aplicação, descrevendo os resultados obtidos, justificações mais completas e conclusões retiradas das diversas componentes.

Capítulo 7 - Conclusões e Trabalho Futuro No último capítulo são descritas todas as conclusões tiradas do trabalho realizado, e proposto trabalho futuro de modo a melhorar o que foi aqui desenvolvido.

Além dos capítulos descritos, são também incluídos vários apêndices de modo a suportar a informação aqui apresentada e discutida, fornecendo uma melhor visualização dos resultados obtidos.

ENQUADRAMENTO

O presente capítulo apresenta um estudo dos temas e potenciais técnicas em que esta dissertação se baseia, apresentando numa fase inicial a temática de Processamento de Imagem seguida pela breve especificação de *Machine Learning* em conjunto com a sua respetiva sub-categoria denominada de classificação como uma potencial técnica de análise colorimétrica. Por fim é também ilustrada a questão da usabilidade no desenvolvimento de interfaces gráficas de sistemas de *software*, em combinação com o seu *Design* e diversas técnicas da sua avaliação.

2.1 Processamento de Imagem

Neste contexto, define-se uma imagem como uma matriz retangular cujos valores são chamados *pixels*, sendo todos do mesmo tipo. Cada *pixel* representa a cor da imagem num dado ponto e é composto pela combinação de um ou mais valores discretos [12].

Com esta definição, pode-se considerar Processamento de Imagem como o conjunto de várias técnicas e operações aplicadas a imagens cujo objetivo é convertê-las para uma outra representação de dados, sendo esta potencialmente outra imagem que esteja num formato utilizável para processamento posterior. Estas operações são muitas vezes aplicadas em várias outras áreas semelhantes como Visão por Computador, Fotografia Computacional e Reconhecimento Computacional, sendo necessário grande cuidado no seu uso de modo a obter resultados aceitáveis [24].

De seguida, irá ser apresentada uma das técnicas mais usadas de Processamento de Imagem para diversos objetivos, seguida de várias características de imagens que é possível extrair, assim como diversas técnicas usadas para esse fim.

2.1.1 Filtros de convolução

Um filtro consiste numa operação chamada de convolução que funciona pela passagem de uma matriz chamada *kernel* por todos os *pixels* de uma imagem, modificando-os com base na sua vizinhança intersetada com a matriz. Ao colocar-se o *kernel* por cima de uma região da imagem, o índice do *pixel* a ser modificado é identificado em relação à matriz e denominado de âncora. Esta matriz pode ter vários tamanhos diferentes, sendo os tamanhos mais comuns 3×3 , 5×5 ou 7×7 e normalmente, nestes *kernels*, a posição da âncora é o índice do centro. Assim, dada uma imagem $I(x, y)$ e um *kernel* $G(i, j)$ de dimensões $M_i \times M_j$ e o ponto de âncora do *kernel* (a_i, a_j) a imagem resultante da convolução $H(x, y)$ de mesmo tamanho é definida pela fórmula:

$$H(x, y) = \sum_{i=0}^{M_i-1} \sum_{j=0}^{M_j-1} I(x + i - a_i, y + j - a_j) G(i, j) \quad (2.1)$$

Estes filtros são comumente usados para reduzir o ruído de uma imagem, isto é, a remoção de artefatos ou formas não desejadas que foram produzidos de forma aleatória. No entanto, existem outros usos diferentes desta técnica como detecção dos contornos de uma imagem (descrito na secção 2.1.3) ou, em outro tipo de filtros como de operações morfológicas, que usam *kernels* especiais para isolar ou juntar elementos de uma imagem [7].

2.1.2 Cor

A cor, na sua definição física consiste na sensação que chega ao olho ou a um dispositivo de captura de imagem a partir de várias ondas de luz de diferentes comprimentos. Estas ondas têm como origem uma fonte de luz ou como resultado da reflexão com um objeto encontrado a meio do percurso, chegando aos olhos ou a um dado sensor diretamente da fonte. As ondas de luz refletidas são, essencialmente a causa da cor que se observa em todos os objetos [12].

A cor pode ser descrita por diversos modelos, uns mais usados que outros e com aplicações diversas.

CIE XYZ Um dos primeiros modelos universais de representação da cor, criado em 1931 pela CIE (Comissão Internacional de Iluminação). Este modelo funciona com base em 3 valores numéricos: X , Y e Z que possuem a capacidade de representar todas as cores que o olho observa através da combinação física de 3 cores (vermelho, verde e azul) capturadas pelos cones, células especiais sensíveis a essas mesmas cores.

De modo a definir graficamente a cor, estas 3 primitivas devem ser convertidas em valores independentes do brilho a partir das fórmulas:

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad z = \frac{Z}{X + Y + Z} \quad (2.2)$$

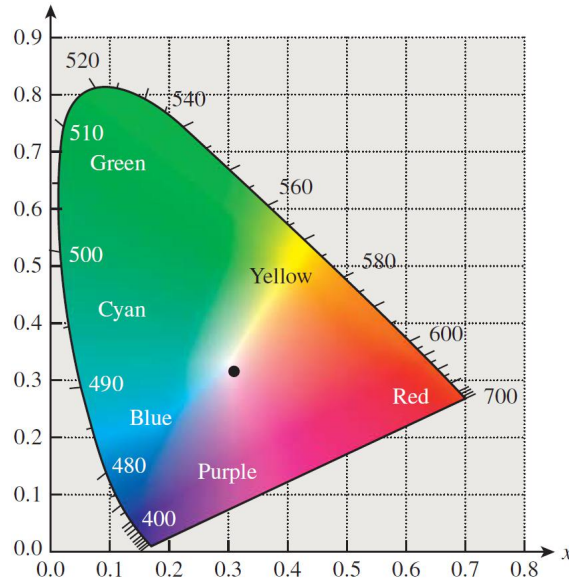


Figura 2.1: Diagrama cromático CIE XYZ. A fronteira do diagrama representa a correspondência da cromaticidade com o comprimento de onda de luz monoespectral em nanômetros. De notar que este corte de $x+y+z = 1$ não representa todas as cores existentes. [12].

Sendo que $x + y + z = 1$. Com estes valores, pode-se definir o *diagrama do espaço cromático CIE XYZ* apenas com base no x e y [12].

RGB Conhecido por ser o modelo de cor aditivo, este é o mais usado na representação de imagens e também em diferentes tipos de ecrãs digitais. Cada cor é representada por 3 componentes denominadas R (Vermelho), G (Verde) e B (Azul), sendo que cada um dos valores das componentes indica a sua influência na cor final. A ausência de qualquer influência de cada componente origina a cor preta, enquanto que a influência equivalente de cada componente na cor origina em tons de cinzento dependentes dos valores até à influência máxima que origina a cor branca. De notar que este modelo de cor não possui a capacidade de representar todas as cores, embora seja possível fazer a conversão da cor para o modelo CIE XYZ pela relação:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 0.18760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.3)$$

De modo a obter-se a conversão de RGB para CIE XYZ, usa-se a matriz inversa. De notar que esta conversão de CIE XYZ para RGB é aproximada para as cores que o modelo não consegue representar [12, 24].

CMY É um modelo de cor subtrativo e é mais usado para descrever cor em impressoras, também representado por 3 componentes de cor denominadas C (Ciano) M (Magenta) Y (Amarelo). Estas componentes comportam-se com base na porção da luz que absorvem, sendo que o Ciano absorve a luz vermelha mas reflete azul e verde,

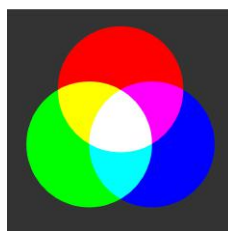


Figura 2.2: Paleta de cores do modelo RGB [24].

o Magenta absorve o verde e o amarelo absorve o azul. Na prática, não é possível combinar estas 3 componentes para formar a cor preta, pois a sua mistura não absorve toda a luz, por isso as impressoras usam um modelo estendido do *CMY* com a adição da cor preta (representada por *K*) chamado *CMYK*. Tal como o modelo *RGB*, os modelos *CMY* e *CMYK* não possuem a capacidade de representar todas as cores possíveis [12].

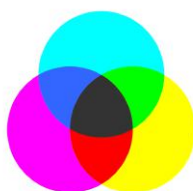


Figura 2.3: Paleta de cores do modelo CMY [24].

HSV Este modelo foi criado com o objetivo de simplificar a seleção de uma determinada cor. Possui 3 componentes: *Hue* que consiste na identificação de qual a cor pura numa roda de diferentes cores, a *Saturation* e o Valor. Existe também um outro modelo semelhante e com o mesmo objetivo chamado *HLS* identificado igualmente por 3 componentes, o *Hue*, a *Lightness* que representa a claridade da cor e a *Saturation* [12].

YUV O modelo de cor *YUV* é um modelo criado para representar a cor em vídeo, nomeadamente para o standard *PAL*, usado na Europa. Composto por 3 componentes: *Y* que representa a luminância que a cor apresenta, e duas outras componentes *U* e *V* que representam a cromaticidade da cor em frequências mais baixas [24].

De seguida efetua-se uma análise a algumas técnicas de extração de informação desejada ou conversão para outro formato a partir da cor.

2.1.2.1 Descritores

Chama-se um descritor de imagem à descrição de uma característica produzida por algoritmos aplicados a uma imagem [20]. Assim, a partir dessas descrições é possível criar sistemas de pesquisa de conteúdos em imagens [1].

De seguida são apresentados alguns exemplos de descritores a partir da cor.

Histograma Um histograma é um descritor que identifica uma imagem a partir da sua distribuição discreta da cor. A forma mais comum deste descritor consiste em, para cada valor possível de cor numa imagem, contar quantos *pixels* existem com essa mesma cor, sendo isto feito para cada componente, independentemente do modelo de cor. No entanto, a partir desta definição é possível criar outras variantes de histogramas, como histogramas locais que consistem em dividir a imagem em vários blocos do mesmo tamanho e calcular os seus histogramas individuais.

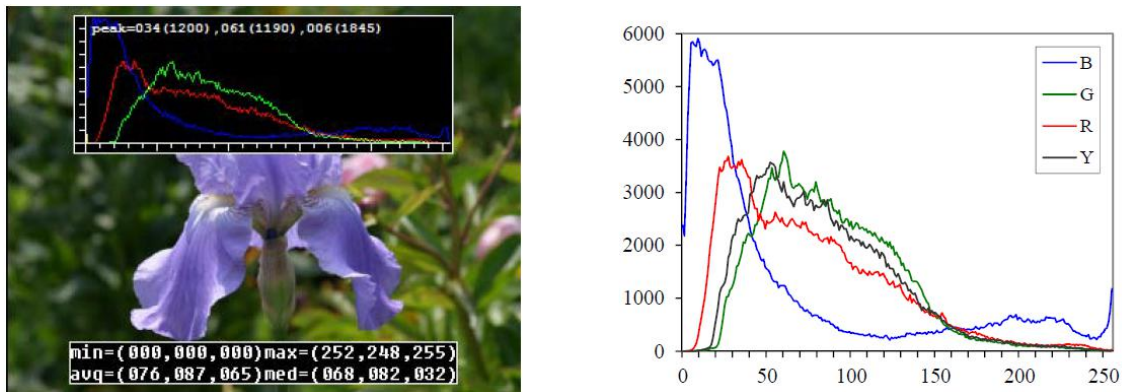


Figura 2.4: Uma imagem e o seu respetivo histograma global, assim como algumas medidas com base nos *pixels*. [24].

A figura 2.4 mostra como exemplo uma imagem de uma flor com o seu histograma por cima e ao lado, o mesmo histograma em questão dos canais de cor *RGB*, assim como a componente *Y* do modelo *CIE XYZ*. De notar que a grande predominância de valores verdes no gráfico deve-se à imagem ser bastante verde, principalmente no fundo [1].

Momentos de cor Os momentos de cor são características da distribuição discreta da cor de uma imagem, sendo usadas normalmente as seguintes: a média, o desvio padrão e a obliquidade. Estas medidas, como os histogramas podem ser calculadas globalmente ou localmente, dividindo também a imagem em vários blocos, sendo calculadas também para cada componente de cor de cada *pixel* [1].

2.1.3 Forma

Um tipo de descritores útil para sistemas de pesquisa de conteúdos em imagens consiste na detecção de formas, sendo ideal para encontrar correspondências entre o que se deseja detetar em imagens [20].

2.1.3.1 Thresholding

Uma das técnicas básicas de detecção de formas consiste no *thresholding*. Esta técnica permite rejeitar ou aceitar *pixels* de uma imagem comparando-os a um valor chamado de *threshold*. Esta operação é normalmente aplicada a imagens cuja cor em cada ponto é representada por um tom de cinzento, embora seja possível aplicar a imagens em outros formatos de cor. Assim, dada uma imagem $img(x, y)$ do tamanho $w \times h$ com $0 \leq x < w$ e $0 \leq y < h$, a imagem resultante da operação de *thresholding* mais geral $res(x, y)$ denominada *binarizada* ou *binária* do tamanho $w \times h$ com $0 \leq x < w$ e $0 \leq y < h$ é dada pela fórmula:

$$res(x, y) = \begin{cases} maxVal & img(x, y) > threshold \\ minVal & cc \end{cases} \quad (2.4)$$

Sendo $minVal$ a cor preta e $maxVal$ a cor branca. A partir desta fórmula é possível usar diferentes variações da operação, como o cálculo dinâmico do *threshold* a aplicar numa determinada região da imagem, chamada *threshold* adaptativo [7].

2.1.3.2 Transformada de Hough

A Transformada de *Hough* é uma técnica usada para encontrar diversas formas simples numa imagem binária. A versão original desta técnica é usada para detetar linhas, tendo sido posteriormente criadas outras variações.

Segundo esta técnica, qualquer ponto não nulo (x_i, y_i) numa imagem binária pode pertencer a uma possível linha. Para isso, para cada um desses pontos calcula-se a representação das possíveis linhas em coordenadas polares pela expressão:

$$\rho = x_i \cos \theta + y_i \sin \theta \quad (2.5)$$

sendo $\rho \geq 0$ e $\theta \in [0, 2\pi]$ em radianos. Com todos os pares (ρ_i, θ_i) calculados para cada ponto, forma-se uma curva no chamado plano acumulador (ρ, θ) . Estas curvas mostram todas as linhas a partir das suas interseções, pois se a curva originada de qualquer ponto da imagem (x_i, y_i) interseja a curva originada de outro qualquer ponto (x_j, y_j) diferente então considera-se que ambos os pontos pertencem à mesma linha [7].

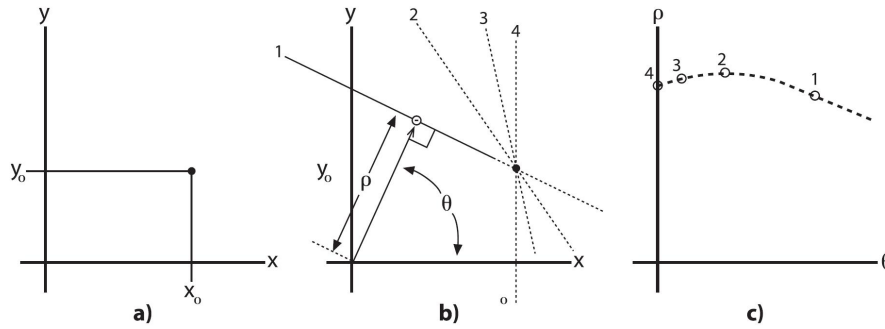


Figura 2.5: a) Um ponto (x_0, y_0) no plano da imagem. b) As diferentes linhas parametrizadas por (ρ_i, θ_i) para o ponto. c) Curva do plano acumulador (ρ, θ) para o ponto da figura a) [7].

2.1.3.3 Filtros de detecção de contornos

Um filtro de detecção de contornos, na sua forma mais geral consiste na aplicação de um filtro com um *kernel* especial que converte a imagem apenas para a representação dos contornos do seu conteúdo. Um exemplo de um destes *kernels* é o operador *Sobel*. Este operador pode ser dividido em dois: o operador de *Sobel* em x , que realça contornos horizontais e o operador de *Sobel* em y que realça contornos verticais, sendo estas operações as derivadas. Este operador no entanto nem sempre é o ideal, pois apresenta apenas resultados bastante aproximados e não muito precisos com imagens consideradas de frequências altas, sendo normalmente usado em combinação com outras operações para obter uma melhor representação de contornos [24].

2.1.3.4 Detetor de Canny

O detetor de *Canny* é o nome dado a um algoritmo usado para detecção de contornos numa imagem, sendo mais sofisticado que apenas um filtro para o mesmo objetivo. Este algoritmo possui quatro passos:

1. Aplicação de um filtro com um *kernel* cujos elementos pertencem à curva *Gaussiana* com o objetivo de remover qualquer eventual ruído que a imagem possa ter.
2. Aplicação de dois filtros semelhantes aos de *Sobel* em x e em y de modo a obter as derivadas da imagem.
3. Com os valores do passo anterior, calcula-se a amplitude do gradiente e a sua respetiva direção do resultado da aplicação dos filtros.
4. Remove-se todos os *pixels* não considerados como parte de um contorno.
5. Realiza-se uma filtragem final dos *pixels* com base em dois *thresholds*, sendo um maior que o outro.

Na primeira filtragem de todos os *pixels*, um *pixel* é removido se o seu gradiente não for um máximo local.

A filtragem final é feita com base nos respetivo gradiente de cada *pixel*, sendo que se um dado gradiente desse *pixel* for superior ao *threshold* de cima, este é aceite como um contorno, se for inferior ao *threshold* de baixo, este é rejeitado e se estiver entre os dois *thresholds* então apenas será aceite se estiver ligado a um outro *pixel* já aceite [7, 17].

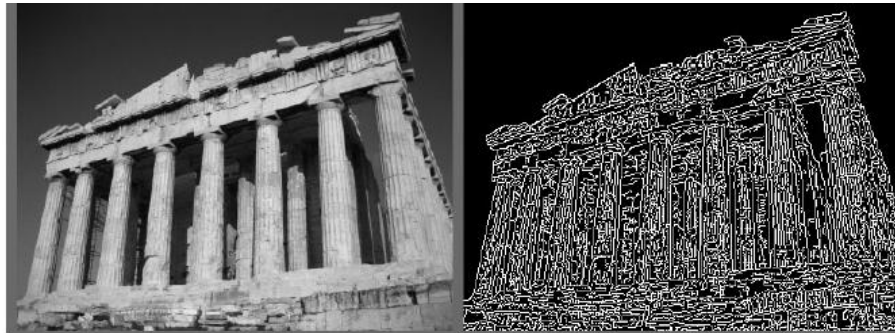


Figura 2.6: Imagem original e resultado da aplicação do algoritmo *Canny* [7].

2.1.3.5 *Hough Circle Transform*

A *Hough Circle Transform* ou transformada de *Hough Circles* é uma variação da transformada original de *Hough* em que, em vez de detetar linhas numa imagem, deteta círculos. Embora esta transformada possa funcionar de maneira semelhante à original, é usada uma técnica chamada Gradiente de *Hough*, de modo a usar menos memória. Esta técnica funciona com base num acumulador, representado por uma matriz bidimensional que é usada para registar as interseções dos pontos da imagem. Este método consiste nos seguintes passos [7]:

1. Aplicação do detetor de *Canny* na imagem.
2. Cálculo dos gradientes da imagem a partir das derivadas do filtro de *Sobel* em x e y .
3. Incremento dos valores do plano acumulador referentes aos pontos de contornos pertencentes às linhas dos gradientes cuja distância esteja num dado intervalo.
4. A partir dos pontos registados no plano, são escolhidos como possíveis centros de círculos na imagem se o seu valor na matriz for superior a um dado *threshold* e ao valor dos seus pontos vizinhos.
5. Todos os centros escolhidos são posteriormente ordenados por ordem decrescente do seu valor na matriz.
6. Para cada possível centro escolhido, ordenam-se todos os outros *pixels* não nulos com base na sua distância ao centro e seleciona-se um raio que seja compatível com o maior número desses *pixels*.

7. Se um dos possíveis centros tiver suficiente suporte de *pixels* não nulos do passo anterior ou, se tiver distância suficiente de um outro centro, esse ponto e o seu raio passam a ser considerados um círculo da imagem.

2.1.4 Classificação

A classificação permite identificar e separar dados em classes distintas e é uma das técnicas usadas na área de *Machine Learning*. Esta área, um campo de Inteligência Artificial, tem ganho grande popularidade na resolução de vários problemas de diversas áreas, cujo potencial se destaca por mostrar resultados cada vez mais precisos com base na aprendizagem. Esta técnica permite resolver um dado problema sem este ser programado explicitamente, sendo apenas necessário "ensinar" um algoritmo como chegar à solução a partir de um conjunto de dados. Este conjunto é constituído por dados semelhantes aos dados que o algoritmo deve processar num cenário real, com a principal diferença que é também fornecido o resultado esperado de cada entrada de dados [2].

A partir da definição, os algoritmos de classificação podem ser divididos em diferentes tipos de problema a resolver [21]:

Binária Neste tipo de classificação, os dados são classificados em uma e só uma classe de apenas duas existentes, sendo este considerado o tipo de problema de classificação mais elementar e comum.

Multi-Classe Neste tipo de classificação, os dados são classificados em uma e só uma classe de l classes existentes, sendo $l > 2$.

Etiquetas múltiplas Ao contrário dos problemas anteriores, em classificação de etiquetas múltiplas, os dados são classificados em mais do que uma das l classes existentes.

Hierárquica Neste tipo de classificação, os dados são classificados em uma e só uma classe de todas as classes existentes, sendo estas divididas em sub-classes ou agrupadas em várias.

Estes conceitos são úteis na resolução de diversos problemas de Processamento de Imagem e Visão por Computador, usando descritores de imagens, seja a partir da cor como se descreve na secção 2.1.2 ou de formas como na secção 2.1.3.

O uso de um algoritmo de *Machine Learning* inclui assim os seguintes passos [7]:

Recolher e etiquetar dados Para treinar um destes algoritmos é necessário recolher dados relevantes e etiquetá-los consoante o resultado desejado, sendo necessário que sejam bastante explícitos e variados de modo a aumentar a precisão.

Considera-se que os algoritmos de classificação pertencem à classe de algoritmos aprendizagem supervisionada, pois para serem usados é necessário etiquetar o conjunto de dados com que se vai treinar. [7].

Extrair descritores Depois da recolha e etiquetação dos dados é necessário extrair descritores relevantes para o que se pretende obter. Este passo depende bastante do problema em questão, sendo esses descritores posteriormente convertidos numa representação numérica dos dados chamada *Feature Vectors* ou apenas *features*. Para estes vetores é associada a sua devida etiqueta relacionada com o resultado que representam.

Uma das técnicas usadas na geração de descritores relevantes consiste numa operação que converte os descritores iniciais de dimensão d para uma dimensão k , sendo $k < d$, com perda mínima de informação chamada de Análise de Componentes Principais (do inglês Principal Component Analysis ou PCA). Além disso, é bastante comum aplicar uma operação de escala em todos os dados, de modo a obter uma normalização, garantindo que todos os dados se encontrem na mesma ordem de grandeza.

Partição dos dados Com todos os dados extraídos e convertidos, estes são particionados em 3 conjuntos diferentes: Um conjunto de treino que serve para ensinar o algoritmo, um conjunto de validação para caso se deseje realizar uma avaliação intermédia dos resultados do algoritmo ou modificar alguns dos seus parâmetros e um conjunto de teste para realizar a avaliação final dos resultados do classificador.

No processo de treino, como boa prática deve-se treinar o algoritmo a partir de técnicas de Validação Cruzada (do inglês Cross Validation). Estas técnicas consistem em dividir o conjunto de dados em K subconjuntos que devem ser executados em diversas sessões de treino. Nestas sessões, são usados diferentes conjuntos de dados com os diversos respetivos papéis de treino, validação e testes, medindo o desempenho do classificador pela média dos resultados do desempenho de cada uma das sessões, sendo que essa medida fornece um valor mais preciso de como o algoritmo final, chamado de modelo se comporta com dados novos [7].

Todos estes passos servem para grande parte de tipos diferentes de algoritmos de *Machine Learning* que realizam classificação. De seguida serão apresentados alguns algoritmos importantes de classificação que podem ser usados em conjunto com técnicas de Processamento de Imagem.

K-Nearest Neighbors Sendo considerado o classificador mais simples, este algoritmo guarda o conjunto de treino com as suas etiquetas como pontos e classifica um novo ponto com base numa maioria de votos dos K pontos de treino de dados mais próximos. É considerado bastante eficiente na classificação, podendo apresentar resultados bastante bons mas também é lento e possui grandes requisitos de memória [7].

Random Trees Este classificador é constituído por um conjunto de várias árvores de decisão. Estas árvores determinam um *threshold* e um descritor de dados no seu nó corrente que melhor divida os dados nas respetivas classes, sendo este procedimento repetido recursivamente. Este conjunto de árvores é construído até uma determinada profundidade ou largura máxima, em que no processo de treino, cada nó de cada árvore determina a sua divisão a partir de um subconjunto aleatório de descritores dos dados, garantindo que cada árvore possa tornar a sua decisão de forma estatisticamente independente. Este classificador é bastante eficiente na maior parte dos diferentes tipos de conjuntos de dados [7].

Support Vector Machines O SVM é um algoritmo que classifica os dados a partir da criação de um *hiperplano* (uma generalização de uma linha para várias dimensões) ótimo que os separa em classes distintas, a partir da resolução de um problema de programação quadrática com restrições de desigualdade. Este *hiperplano* ótimo consiste, além da separação correta dos dados em classes, na maximização da distância da sua margem com os dados.

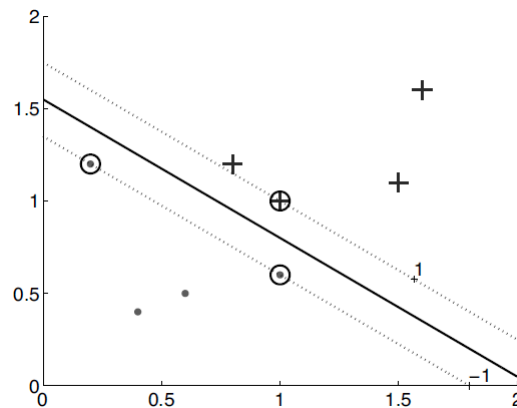


Figura 2.7: Visualização gráfica de uma classificação realizada com SVM de duas classes de dados distintos em cada lado do *hiperplano* [2].

Least Squares Support Vector Machines O LS-SVM é idêntico ao SVM sendo apenas uma formulação do mesmo, com a diferença que os dados são classificados a partir da resolução de equações lineares com restrições de igualdade. A principal vantagem deste classificador é o facto que, por classificar dados a partir da resolução de equações lineares, apresenta custos de memória e de processamento mais baixos comparados com o SVM normal [23].

2.1.4.1 Avaliação de Classificação

A correção da classificação pode ser avaliada pela determinação de: O número de exemplos de uma classe corretamente identificados (chamados de verdadeiros positivos ou do inglês *true positives*), o número de exemplos corretamente identificados que não pertencem a uma classe (verdadeiros negativos do inglês *true negatives*), O número de exemplos que foram incorretamente atribuídos a uma classe (falsos positivos do inglês *false positives*) e o número de exemplos de exemplos que não foram reconhecidos para nenhuma classe (falsos negativos do inglês *false positives*). A partir destes dados, é possível, para classificação arranjar 3 métricas fundamentais, sendo que para classificação multi-classe, se divide cada uma das métricas em 2 categorias diferentes: *Micro* (μ), que favorece classes com mais dados e *Macro* (M) que trata todas as classes de igual forma. Assim, descrevendo as 3 métricas e as suas respetivas fórmulas [21]:

Exatidão (*Accuracy*) A exatidão é a métrica mais básica de medição de desempenho de um classificador, sendo apenas calculada pela fórmula:

$$A = \frac{tp + tn}{tp + fn + fp + tn} \quad (2.6)$$

Sendo tp os verdadeiros positivos, tn os verdadeiros negativos e fp e fn os falsos positivos e falsos negativos respetivamente. Em classificação multi-classe, esta métrica é calculada com base na fórmula:

$$A = \frac{\sum_{i=1}^l \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l} \quad (2.7)$$

Sendo l o número de classes, representando assim a exatidão média do modelo com base em todas as classes.

Precisão (*Precision*) A precisão é uma métrica dada com base nos verdadeiros positivos e nos falsos positivos, e em classificação binária é representada pela fórmula:

$$P = \frac{tp}{tp + fp} \quad (2.8)$$

Sendo tp os verdadeiros positivos e fp os falsos positivos. Em classificação multi-classe, define-se as duas categorias *Micro* e *Macro* pelas respetivas fórmulas:

$$P_{\mu} = \frac{\sum_{i=1}^l tp_i}{\sum_{i=0}^l tp_i + fp_i} \quad (2.9)$$

$$P_M = \frac{\sum_{i=1}^l \frac{tp_i}{tp_i + fp_i}}{l} \quad (2.10)$$

Sensibilidade (*Recall*) A sensibilidade consiste na eficácia que um classificador tem a identificar corretamente classes positivas, representada pela fórmula em classificação binária:

$$R = \frac{tp}{tp + fn} \quad (2.11)$$

Sendo tp os verdadeiros positivos e fn os falsos negativos. Em classificação multi-classe, dividindo da mesma forma em *Micro* e *Macro* respetivamente:

$$R_{\mu} = \frac{\sum_{i=1}^l tp_i}{\sum_{i=1}^l (tp_i + fp_i)} \quad (2.12)$$

$$R_M = \frac{\sum_{i=1}^l \frac{tp_i}{tp_i + fp_i}}{l} \quad (2.13)$$

FScore O *FScore* é uma métrica que identifica a relação entre as etiquetas positivas e os outros resultados dados pelo classificador. A fórmula para classificação binária:

$$FScore = \frac{(\beta^2 + 1)tp}{(\beta^2 + 1)tp + \beta^2 fn + fp} \quad (2.14)$$

Com as fórmulas para *Micro* e *Macro* respetivamente:

$$FScore_{\mu} = \frac{(\beta^2 + 1)P_{\mu}R_{\mu}}{\beta^2 P_{\mu} + R_{\mu}} \quad (2.15)$$

$$FScore_M = \frac{(\beta^2 + 1)P_MR_M}{\beta^2 P_M + R_M} \quad (2.16)$$

Sendo β um parâmetro para a fórmula e $\beta = 1$ um dos valores mais usados.

2.2 Detecção e Quantificação Colorimétrica

A quantificação colorimétrica (ou análise colorimétrica) consiste no processo de leitura de valores tirados de biossensores ou substâncias químicas com base na cor que apresentam, sendo este o passo em que se converte essa mesma cor para um dado valor.

No contexto desta dissertação, pretende-se usar um modelo de quantificação a partir da cor de uma determinada zona de uma imagem que contenha um biossensor. Este modelo requer um uso cuidadoso da luz ambiente, temperatura e distância do dispositivo de captura (normalmente uma câmara) com a amostra a medir de modo a obter valores mais corretos [19].

2.2.1 Quantificação a partir de *smartphones*

A partir da câmara, um *smartphone* possui um grande potencial como dispositivo de quantificação colorimétrica, sendo capaz de reconhecer diferenças mínimas na tonalidade da cor. Esta capacidade é baseada na lei de *Beer-Lambert* que relaciona a concentração de uma dada substância química com a intensidade da cor que esta apresenta.

A maior parte das aplicações desenvolvidas com este objetivo usa o modelo *RGB* de modo a representar a cor de uma imagem, sendo que cada um dos canais deste modelo possui valores no intervalo $[0, 255]$, sendo assim possível distinguir $256^3 = 16777216$ cores diferentes [19].

2.3 Usabilidade

A Usabilidade é uma propriedade importante a ter em conta no desenvolvimento de qualquer sistema de *software*, mais precisamente da sua interface gráfica. Esta propriedade possui várias características que a definem, sendo essas [16]:

Aprendizagem O sistema deve ser fácil de se aprender a usar de modo a que o utilizador possa ser produtivo.

Eficiência O sistema deve ser eficiente de modo a que, assim que um utilizador o aprenda a usar, este possa ser altamente produtivo.

Memorabilidade O sistema deve ser fácil de se lembrar de modo a que um utilizador casual seja capaz de voltar a usá-lo facilmente sem precisar de aprender tudo novamente.

Erros O sistema deve ter um rácio baixo de erros, de forma a que os utilizadores cometam poucos erros no seu uso e, caso aconteça, que possam corrigi-los facilmente.

Satisfação O sistema deve ser agradável de se usar de modo a satisfazer os utilizadores.

De modo a avaliar estes requisitos, existem várias técnicas possíveis, que irão ser apresentadas de seguida.

2.3.1 Avaliação Heurística

A avaliação heurística é uma técnica de avaliação realizada por um pequeno conjunto de peritos que examinam a interface e a avaliam, tentando arranjar uma opinião acerca do que tem de bom e de mau. Idealmente isto deve ser feito com base em regras específicas e neste contexto tentar encontrar problemas de usabilidade que uma interface gráfica possa ter, de modo a poderem ser resolvidos num processo iterativo de *design* [16].

2.3.2 Questionários de Usabilidade

Os questionários são outro método de avaliação de Usabilidade, mas no contexto dos utilizadores do sistema. Este método é baseado no conceito que vários aspetos da usabilidade podem ser estudados por simplesmente perguntar aos utilizadores a partir de questionários de resposta fechada. Num questionário, os utilizadores devem fornecer respostas factuais da sua interação com o sistema, normalmente assinalando as tarefas que mais realizam ou definir a sua opinião com base numa escala fornecida [16].

De seguida é apresentado um exemplo de questionário.

2.3.2.1 *Computer System Usability*

Este tipo de questionário tem como principal propósito avaliar a satisfação do utilizador com a usabilidade de um sistema em cenário de uso real. Consiste em 19 perguntas de resposta fechada, que o utilizador deve responder numa escala o quanto concorda sobre o que sentiu na utilização do sistema [14].

Alguns exemplos de questões:

1. Estou satisfeito com a facilidade de uso do sistema
2. Este sistema é simples de se usar
3. Sou capaz de realizar o meu trabalho rapidamente
4. Sinto-me confortável a usar o sistema
5. Quando me engano a usar o sistema, consigo resolver rapidamente
6. A informação proveniente do sistema é fácil de se entender
7. O sistema possui todas as funcionalidades e capacidades que esperava

2.3.3 *Design Gráfico*

O *Design Gráfico* é um elemento importante de modo a criar um diálogo natural com interfaces gráficas de sistemas de *software*. É constituído por diferentes princípios (*guidelines*) com o objetivo de ajudar os utilizadores a priorizarem a sua atenção aos elementos mais importantes da interface.

Enunciam-se assim alguns princípios para o *design* de interfaces [16]:

Não fazer demasiado Uma interface não deve ser constituída por uma grande mistura de cores altamente contrastantes de saturação elevada. É preferível limitar a um número pequeno de cores predominantes pelo *design*. Cores cinzas claros, ou em tons pastel são ideais como cores de fundo.

Certificar que a interface é utilizável sem cor Cerca de 8% das pessoas do género masculino possuem daltonismo, o que significa que de modo a tornar o sistema utilizável para todo o tipo de pessoas, deve-se adicionar informação suplementar de modo a não deixar que existam componentes da interface unicamente identificáveis pela sua cor.

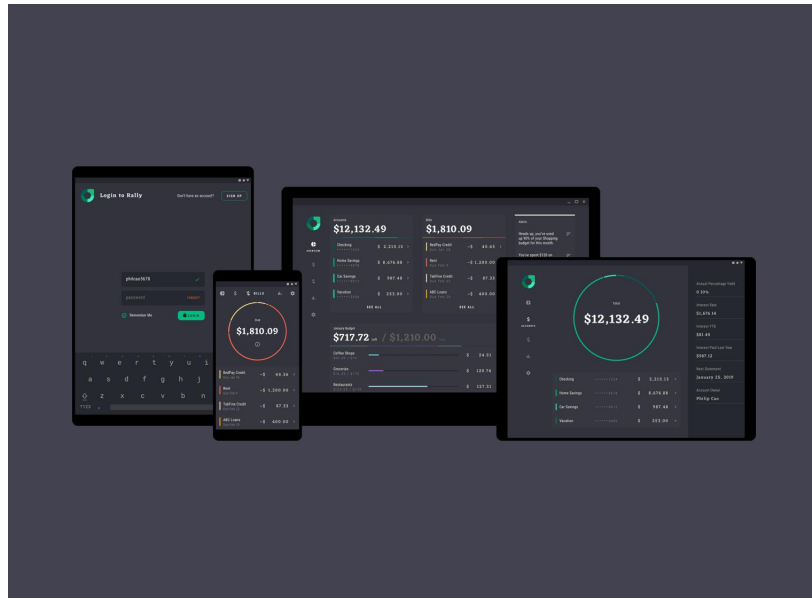
Uso cuidadoso da cor Deve-se usar a cor apenas para categorizar, diferenciar ou destacar e não para fornecer informação, principalmente informação quantitativa.

De seguida apresenta-se um exemplo moderno de um conjunto de princípios de *Design* de interfaces gráficas.

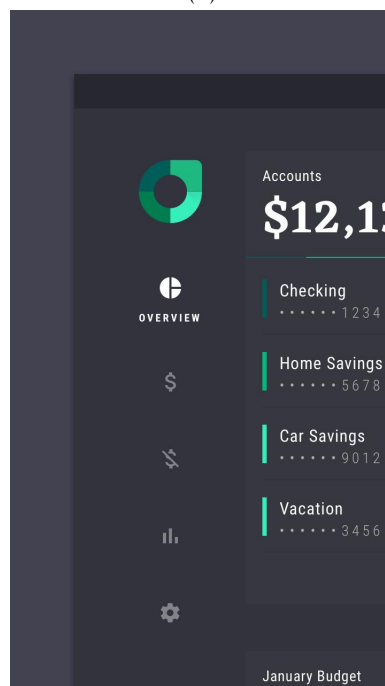
2.3.3.1 *Material Design*

O *Material Design* é uma linguagem visual desenvolvida pela *Google* com objetivos de sintetizar princípios de bom *design* e desenvolver um sistema que unifique a experiência de utilizador em diversas plataformas e diferentes dispositivos [8]. Esta linguagem visual baseia-se na reflexão de luz e nos sombreamentos das diferentes texturas do mundo físico, sendo bastante usada na grande maioria das aplicações desenvolvidas para o sistema operativo *Android*, principalmente criadas pela *Google* [8].

Na figura 2.8 pode-se observar um exemplo prático do resultado das diferentes regras do *Material Design* numa aplicação já existente para gestão de finanças pessoais, chamada *Rally*.



(a)



(b)

Figura 2.8: Exemplo de uso de *Material Design* em *Rally*, uma aplicação de gestão de finanças pessoais [9].

TRABALHO RELACIONADO

Neste capítulo irá ser apresentado o estudo do trabalho relacionado com os temas e objetivos desta dissertação, descrevendo assim alguns exemplos de aplicações que tiram uso da tecnologia de processamento de imagem, realçando no entanto as respetivas técnicas simples e avançadas de quantificação a partir de imagens de biossensores e de ferramentas de análises concebidas para papel e capturadas por diferentes câmaras de dispositivos móveis.

3.1 Análise de Suor

Um dos primeiros trabalhos a apresentar consiste num biossensor autocolante de análise do suor da pele [13]. Este dispositivo tem como funcionalidades a medição da taxa de suor libertado pela pessoa e a realização de análises químicas a essa mesma substância. Estas análises químicas permitem medir o pH e as concentrações de cloreto, ácido láctico e glucose, sendo realizadas por um *smartphone* ou *tablet* a partir da captura de fotografias do sensor pela câmara.

A figura 3.1 mostra o biossensor com a zona de armazenamento do suor e as 4 diferentes zonas que representam as concentrações das 3 substâncias e o valor de pH. Cada uma das zonas apresenta cores diferentes e em diversos tons, representando uma determinada concentração.

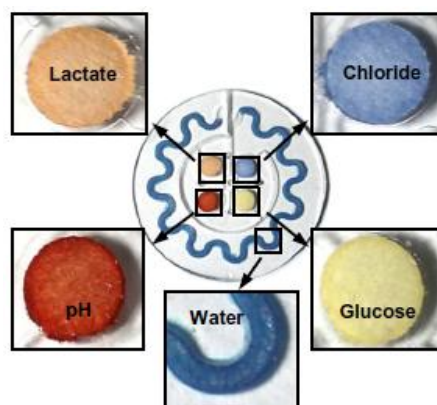


Figura 3.1: Biossensor de captura de suor e as suas respetivas zonas de cor [13].

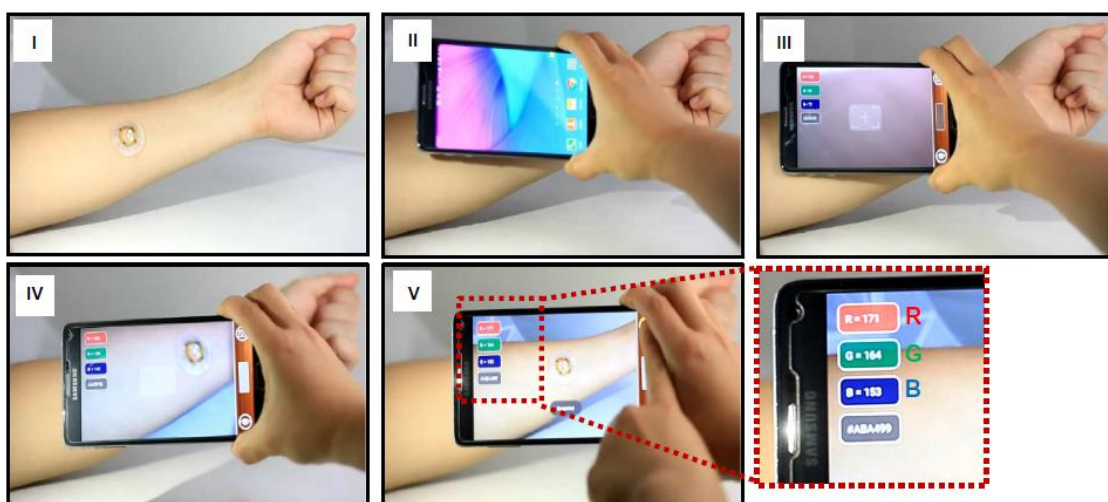


Figura 3.2: Processo de captura de valores periódicos pelo dispositivo móvel [13].

3.1.1 Captura da Imagem do Biossensor

De modo a calcular as diferenças de concentração de cada substância, o rácio de suor e o pH, o dispositivo captura periodicamente imagens do biossensor de modo a observar as diferenças de cor. Este processo é feito com suporte do *NFC* (do inglês Near Field Communication) do dispositivo móvel, que tem como tarefa detetar a distância do dispositivo móvel ao sensor, sendo que este possui a capacidade de comunicar com o dispositivo. Quando o dispositivo se encontra na distância considerada ideal, a aplicação inicia a captura periódica das imagens a partir da câmara.

Cada região de interesse de cada imagem capturada é representada no formato *RGB*, sendo essa cor representada no ecrã do dispositivo caso o utilizador deseje visualizar os valores atuais capturados [13]. Este processo pode ser observado na figura 3.2.

3.1.2 Quantificação Colorimétrica da Análise do Suor

O processo de quantificação colorimétrica consiste numa fase inicial de equilíbrio das cores brancas e pretas em cada imagem capturada com as respectivas referências das cores (preta e branca) encontradas no centro do biossensor, de modo não permitir que a iluminação ambiente ou possíveis sombras afetem os resultados quantificados, seguido pela conversão dos valores de cor *RGB* para valores *RGB%*. Após esta fase, os valores de cor das regiões de interesse são quantificados para os valores desejados a partir de curvas de calibração pré-calculadas [13].

Estas curvas de calibração consistem em gráficos que relacionam a cor com a respetiva concentração, sendo assim possível indicar qual a concentração de uma dada substância com a cor que apresenta.

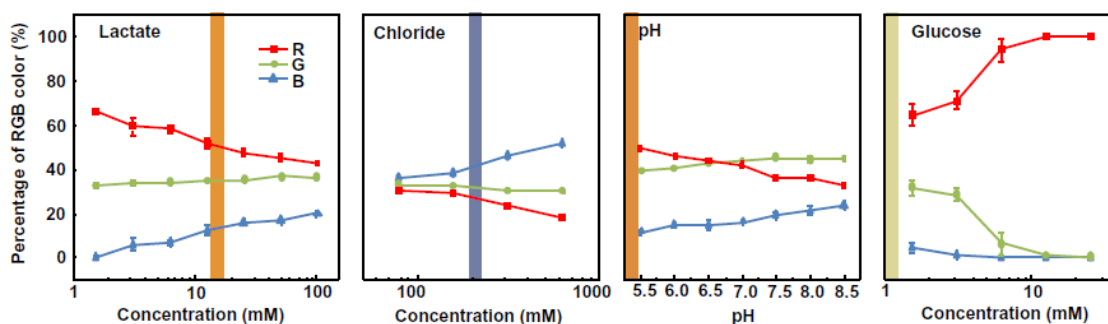


Figura 3.3: Curvas de calibração de cada concentração e pH a calcular. Cada linha representa o valor do canal de cor no eixo das ordenadas e as abcissas o valor a calcular da substância [13].

3.1.3 Resultados

De modo a testar o biossensor e a precisão das suas medidas, foram realizadas duas experiências. A primeira experiência envolveu 9 sujeitos em exercício físico de intensidade média num ambiente controlado com este sensor colocado primeiro nas costas e depois no antebraço e possuindo duas áreas de colheita de suor de tamanhos diferentes.

A segunda experiência foi realizada em contexto competitivo numa corrida de longa distância de ciclismo (*El Tour de Tucson*) realizada por 12 atletas voluntários de modo a verificar a integridade do dispositivo nesse tipo de ambiente.

Os resultados mostraram grandes variações do rácio de suor entre os sujeitos como é suposto, assim como que foi possível medir e distinguir pelas imagens capturadas diferenças de 0.5 unidades de pH ou diferenças de 0.3 a 0.1 mM nas concentrações dos compostos, funcionando corretamente como esperado [13].

3.2 Análise de Urina

Um outro exemplo de uso da técnica de quantificação colorimétrica a apresentar pertence a uma aplicação móvel usada para a leitura dos valores de sensores de testes comerciais de urina. Esta técnica permite medir pH e as concentrações de proteína e glucose a partir da imagem capturada [26].

De modo a realizar a leitura do valor de um dos sensores, o utilizador deve proceder a um processo de calibração manual, que consiste na criação da curva de calibração para um dado teste a uma dada luz ambiente e distância do sensor. Após a conclusão da calibração, os seus dados são armazenados em memória persistente para uso futuro. De notar também que as calibrações apenas podem ser usadas para o mesmo tipo de sensor, na mesma distância e mesma luz ambiente com que foram criadas [26].

De seguida apresentam-se detalhadamente os respetivos procedimentos de calibração e de leitura.

3.2.1 Calibração

O passo de calibração inicia-se pela escolha das opções de tipo de sensor, as substâncias químicas em questão a quantificar, as unidades de concentração e o número de pontos de calibração desejados. Após a introdução destas informações, inicia-se a captura das zonas de referência de calibração com a câmara. Neste processo, para cada secção de cores de referência localizadas nas imagens capturadas é efetuada a média e a transformação dos dados capturados pelos sensores da câmara para valores RGB não lineares $R_c G_c B_c$, seguida pela transformação desses valores para valores lineares $R_l G_l B_l$ a partir das fórmulas:

$$R_l = \left(\frac{0.055 + R_c}{1.055} \right)^{2.4} \quad G_l = \left(\frac{0.055 + G_c}{1.055} \right)^{2.4} \quad B_l = \left(\frac{0.055 + B_c}{1.055} \right)^{2.4} \quad (3.1)$$

De seguida, estes valores são convertidos para valores do espaço de cor $CIE XYZ$ pela relação da equação 2.3.

Para terminar a calibração, estes últimos valores calculados são convertidos para valores do diagrama cromático $CIE XYZ$ pelas equações 2.2. Após calcular estes valores para o número diferente de pontos de calibração desejados e introduzir-se também as respetivas concentrações que cada um dos pontos representa, a aplicação guarda todos os valores numa base de dados interna como curva de calibração para aquela configuração [26].

3.2.2 Análise Colorimétrica do Biosensor

De modo a ler os valores do biossensor, o utilizador deve capturá-lo com a câmara nas mesmas condições de distância, iluminação e temperatura que na respetiva fase de calibração.

Com a imagem capturada, são realizados exatamente os mesmos cálculos que foram feitos na fase de calibração de modo a converter a cor capturada para o diagrama cromático $CIE XYZ$, seguido do primeiro passo extra que consiste no cálculo das distâncias mais

curtas d_k entre o ponto a quantificar com todos os pontos da respetiva curva de calibração (x_k, y_k) previamente calculados pela fórmula:

$$d_k = \sqrt{(x_k - x)^2 + (y_k - y)^2} \quad (3.2)$$

Com k entre 1 e o número de pontos da curva de calibração.

Sabendo os dois valores mínimos d_{ks} e d_{kss} do conjunto de distâncias mais curtas calculadas pela fórmula 3.2, guarda-se $(x_{ks}, y_{ks}), (x_{kss}, y_{kss})$ juntamente com as suas concentrações C_{ks} e C_{kss} e é calculada a distância d_C entre as duas concentrações pela fórmula:

$$d_C = |C_{ks} - C_{kss}| \quad (3.3)$$

De seguida, a aplicação calcula também a distância mais curta d_{sd} entre o ponto de medida à linha entre os dois pontos de calibração pela fórmula:

$$d_{sd} = \frac{|(x_{kss} - x_{ks})(y_{ks} - y_m) - (x_{kss} - x_m)(y_{ks} - y_{kss})|}{\sqrt{(x_{kss} - x_{ks})^2 + (y_{kss} - y_{ks})^2}} \quad (3.4)$$

sendo (x_m, y_m) o valor que se deseja quantificar capturado e convertido no espaço cromático CIE XYZ [26].

A distância calculada na equação 3.3 é usada no cálculo da concentração final a ser medida C_m e no cálculo da variação $v \pm$ do intervalo de concentração d_c dados pelas fórmulas:

$$C_m = \frac{d_p d_c}{d_{xy}} \quad v \pm \frac{d_m d_c}{d_{xy}} \quad (3.5)$$

em que d_m consiste na distância maior entre d_{ks}, d_{kss} e d_{sd} .

Já d_{xy} consiste na distância euclidiana entre os pontos (x_{ks}, y_{ks}) e (x_{kss}, y_{kss}) e d_p a distância entre d_{ks} e d_{sd} dadas pelas respetivas fórmulas [26]:

$$d_{xy} = \sqrt{(x_{ks} - x_{kss})^2 + (y_{ks} - y_{kss})^2} \quad d_p = \sqrt{d_{ks}^2 - d_{sd}^2} \quad (3.6)$$

Na figura 3.4 é possível ver-se graficamente o processo de análise com base na curva de calibração representada.

3.2.3 Resultados

Com este algoritmo, foram realizados testes em *smartphones* considerados de topo e média gama. Os testes colorimétricos foram mergulhados em urina artificial, criada de modo a ter um ambiente controlado e de melhor precisão na confirmação dos valores obtidos. Após a calibração manual para estes testes com os valores de referência incluídos nas embalagens num ambiente com luz fluorescente, foi possível observar resultados com uma precisão elevada, como se pode ver na figura 3.5 [26].

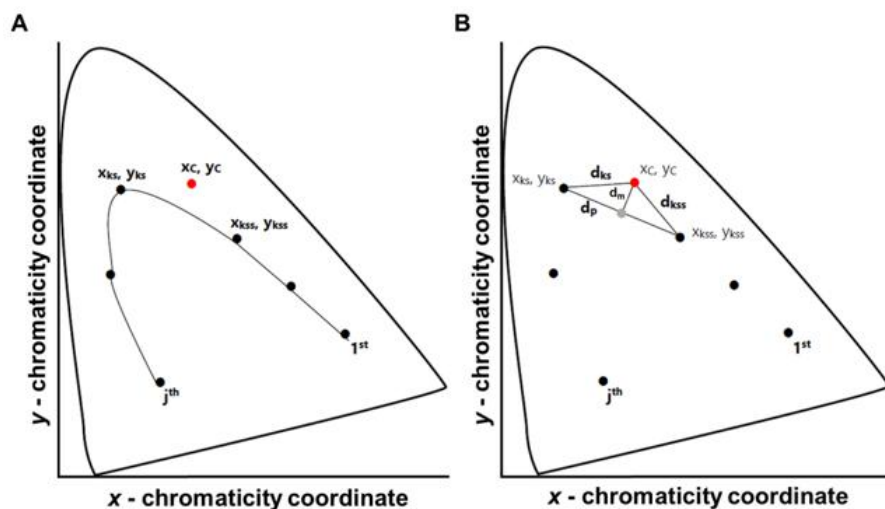


Figura 3.4: Representação de alguns valores calculados e da curva de calibração no gráfico cromático 2D [26].

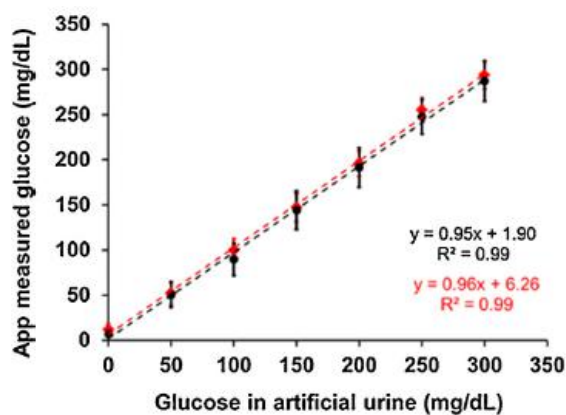


Figura 3.5: Comparação dos resultados medidos pelos dois tipos diferentes de *smartphones* com o seu respetivo erro [26].

3.3 Deteção Colorimétrica com *Machine Learning*

Esta técnica de análise colorimétrica consiste na criação de modelos de *Machine Learning* com os classificadores *SVM* e *LS-SVM* descritos na secção 2.1.4 para o cálculo de valores de pH de imagens de tiras representativas desse valor pela cor e capturadas a partir da câmara de um *smartphone* em vários formatos de imagem diferentes [15].

De seguida são apresentados os passos deste procedimento, assim como os resultados apresentados.

3.3.1 Design Experimental

O primeiro passo consiste na preparação da captura das tiras de pH, criando dois tipos de conjuntos de experiência com iluminação controlada.

O primeiro conjunto de experiências é preparado e capturado num ambiente sem qualquer iluminação além do *flash* do dispositivo móvel que é realizado com base num acessório de impressão 3D para *smartphone*. Já o segundo conjunto é preparado sem *flash* e capturado com diferentes conjuntos de uma e duas fontes de luz simultâneas (lâmpadas fluorescentes e luz solar, lâmpadas de halogéneo e luz solar e lâmpadas fluorescentes e de halogéneo). No primeiro conjunto de experiências e no segundo conjunto com apenas uma fonte de iluminação, foram usadas tiras de pH de 0 a 14.0 em grupos de 4 e com todas as tiras com o mesmo valor em orientações diferentes de modo a ser possível observar as diferenças de iluminação devido à posição em relação ao *flash*. No segundo conjunto com duas fontes de iluminação foram apenas usadas tiras de pH de 4.0 a 9.0 também em várias orientações e posicionadas aleatoriamente, sendo estas posições determinadas por um *script* na linguagem de programação *Matlab* de modo a poder aumentar a variação da construção destes conjuntos de treino e teste [15].

Por fim, também neste passo são usadas tiras de pH nas mesmas condições que os dois conjuntos iniciais, para testes extra cujos valores possuem casas decimais.

3.3.2 Captura de Imagens

A captura das imagens dos conjuntos de experiência foi feita manualmente com o *smartphone* LG G4, sempre com as mesmas definições de configuração da respetiva câmara. No primeiro conjunto de experiências, cada grupo de tiras de pH foi capturado 5 vezes em 6 orientações diferentes, enquanto que no outro conjunto, cada grupo de tiras foi apenas capturado uma vez. Todas as imagens capturadas foram guardadas nos formatos *JPEG* e *RAW*, capturando cerca de 450 imagens em cada um dos formatos de ficheiros para a primeira experiência e cerca de 270 imagens do segundo conjunto [15].

3.3.3 Pré-Processamento

Esta fase consiste na preparação das imagens capturadas para a fase seguinte de extração de características. Neste passo são geradas matrizes de cor de tamanho fixo 700×100 da orientação das imagens de origem. No entanto, antes desta transformação é necessário processar as imagens *RAW*, pois estas consistem apenas em dados puros capturados pelos sensores da câmara. Estas imagens são processadas inicialmente, convertendo-as no formato de imagem *TIFF* que de seguida são transformadas com ajustes de cores e de tons brancos, obtendo assim imagens denominadas *RAW-corrected* ou *RAWc*. O processo de ajuste de cores consiste no uso de *X-Rite ColorChecker Passport* [25], um cartão com diversas cores de referência já previamente determinadas, seguido pela transformação no espaço de cor *CIE XYZ*, voltando no fim ao formato do modelo de cor *RGB*.

Após este processamento, todas as imagens das tiras são colocadas em posição vertical com as respectivas tiras recortadas. De modo a evitar o efeito de esbatimento das pontas das tiras, as zonas do interior são recortadas e colocadas por cima dessas zonas [15].

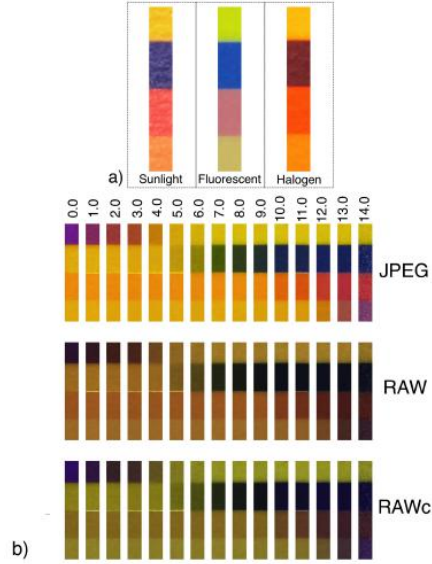


Figura 3.6: a) Influência de iluminações diferentes nas cores numa tira de cor de pH 12.0. b) Diferentes tiras para valores diferentes de pH capturadas em diferentes formatos de imagem com o dispositivo de impressão 3D [15].

3.3.4 Extração de Características

A fase de extração de características consiste na criação de 4 painéis de dados com cores distintas para cada valor de pH nos diferentes formatos de imagem *JPEG*, *RAW* e *RAWc*. Para isso, extraem-se os momentos de cor de cada canal *RGB*, mais precisamente a média, para a criação de uma matriz $4 \times 3 X_n(i, j)$ para o painel de teste i e para a cor j para cada imagem n . Cada uma destas matrizes é então mapeada para um vetor de dimensão 12×1 x_n que será usado para o conjunto de treino do algoritmo. Estes vetores são devidamente etiquetados de modo a existirem valores de treino para os 15 diferentes valores de pH.

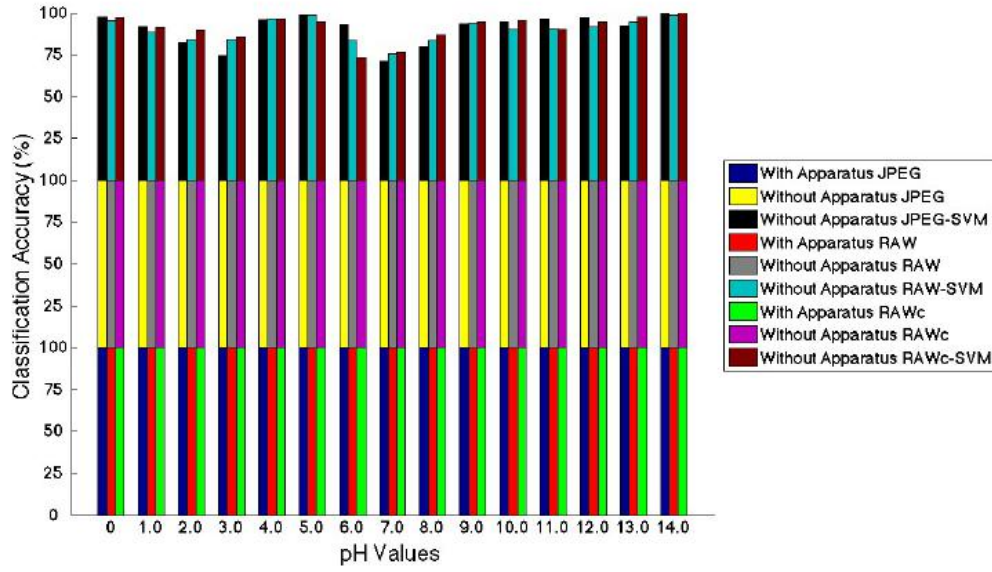


Figura 3.7: Resultados de precisão da detecção automática com e sem isolamento de luz ambiente [15].

3.3.4.1 Classificação dos Dados

De seguida, cada vetor \mathbf{x}_n é classificado pelo SVM e pelo LS-SVM, sendo treinados com um sub-conjunto do conjunto total de treino e depois validados com o conjunto complementar de teste, pela técnica de *Cross-Validation*, com $K = 10$, sendo este um valor de K bastante comum [15].

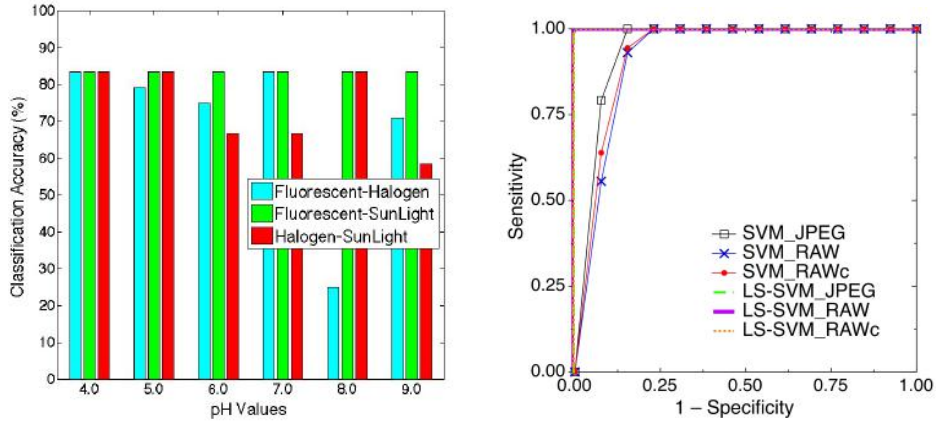
3.3.4.2 Resultados

O desempenho e avaliação dos resultados desta técnica de quantificação foram determinados a partir do cálculo de *Accuracy* (ACC), *Sensitivity* ou *Recall* (SEN) definidas pelas fórmulas na secção 2.1.4.1 e *Specificity* (SPC) definido pela fórmula:

$$SPC = 1 - Recall \quad (3.7)$$

As fórmulas de *Sensitivity* e *Specificity* medem o desempenho do classificador com base no número de resultados corretamente e incorretamente identificados, já a *Accuracy* mede a precisão do classificador a identificar corretamente os valores de pH.

Pela figura 3.7 pode-se verificar que todos os testes de LS-SVM com o dispositivo de impressão 3D, com qualquer formato de imagem testado possuem precisão de 100%. Isto deve-se ao facto do dispositivo de impressão 3D isolar todas as fontes de luz externas que poderiam influenciar a cor. Já o mesmo classificador apresentou uma precisão de aproximadamente 100% nas experiências sem o dispositivo auxiliar. Pela mesma figura, também se pode observar que o formato das imagens não é algo condicionante para os resultados, mostrando que imagens no formato JPEG podem apresentar resultados melhores que imagens no formato RAW [15].



(a) Precisão de classificação do LS-SVM com imagens JPEG com diferentes tipos de fontes de luz

(b) Curvas ROC sem o dispositivo de isolamento de luz dos classificadores.

Figura 3.8: Representação dos resultados dos testes do classificador [15].

Como a *Sensitivity* e *Specificity* dependem de um valor de *threshold* dado para se considerar testes positivos e negativos, foi determinada uma curva ROC (do inglês *Receiver Operating Characteristic*) que mostra a relação entre a *Sensitivity* e $1 - \text{Specificity}$, considerando-se que um bom teste de classificação possui a área por baixo da curva (denominada *AUC*) o mais próximo de 1, sendo $AUC = 1$ considerada uma classificação perfeita. A figura 3.8b mostra as curvas ROC sem isolamento de luz para ambos os classificadores. Pode-se verificar que $AUC = 1$ para todos os formatos de imagem usados para o classificador LS-SVM, enquanto que o classificador SVM obteve $AUC = 0.9729$ para o formato JPEG, $AUC = 0.9525$ para RAW e $AUC = 0.9606$ para RAWc.

No entanto, verificou-se que ao tentar quantificar valores de pH com valores decimais, a classificação é arredondada, não mostrando grande precisão em valores decimais. Este resultado pode no entanto ser contornado adicionando ao conjunto de treino fornecido, tiras de pH com os devidos valores decimais.

Com estes resultados, pode-se também concluir que é possível, além de se obter bons resultados com os clasificadores apropriados e melhorar os resultados de classificação sem um grande gasto de memória com outros formatos de imagem, usando assim o formato JPEG [15].

3.4 Avaliação Crítica

Todas as técnicas e tecnologias descritas neste capítulo e no capítulo anterior formam um conjunto indispensável como estado de arte desta dissertação. Começando pelas técnicas de Processamento de Imagem de detecção de formas e extração de descritores a partir da cor, que formaram um conjunto de ferramentas importantes para o desenvolvimento da aplicação. A técnica de classificação a partir do exemplo descrito na secção 3.3 mostra grande potencial para efetuar a análise colorimétrica, um dos principais desafios desta dissertação.

Além do uso exclusivo desta técnica de classificação, é também possível o uso de técnicas mais simples como as apresentadas na secção 3.1 com base em curvas de calibração, e na secção 3.2, mais precisamente o uso de cores de referência para calibrar a câmara, não permitindo que a iluminação e outras influências externas afetem os resultados. De notar que se pretende que este processo seja automático, ao contrário do que acontece no algoritmo da secção 3.2, tendo sido usada uma tecnologia idêntica à do *X-Rite ColorChecker Passport* [25].

Concluindo este capítulo, a avaliação da Usabilidade da aplicação pôde tirar partido de várias técnicas mencionadas na secção 2.3. Sendo esta uma aplicação com funcionalidades simples do ponto de vista de uso do sistema, pôde-se considerar uma avaliação heurística facilmente realizável assim como a avaliação a partir de questionários por parte de potenciais utilizadores, isto após estes interagirem com a aplicação.

DESENHO DA APLICAÇÃO

Neste capítulo é apresentado o desenho da aplicação, sendo este centrado nas funcionalidades que os utilizadores poderão usufruir. As funcionalidades foram definidas por um *user story* com os respetivos cenários, de modo a demonstrar as possíveis interações com o *software*. Por fim, é também descrita a arquitetura e as tecnologias usadas, apresentando as vantagens nas decisões tomadas de desenvolvimento.

4.1 Funcionalidades e Cenários

A aplicação possui como principais funcionalidades a deteção do biossensor a partir da câmara, seguida pela sua devida análise colorimétrica, devolvendo um resultado ao utilizador. Com base nestas duas funcionalidades, foi desenvolvido o seguinte *user story* de modo a descrever o que o utilizador pode fazer com a aplicação:

Como utilizador, quero medir os meus diabetes de modo a poder controlar esta doença de que eu possa ou não sofrer.

De notar que além deste *user story* é possível desenvolver outras formas de interagir com o sistema de modo a criar sistemas mais complexos e utilizáveis num contexto médico. Assim, foram criados os seguintes cenários, com o objetivo de descrever como o utilizador interage com a aplicação de modo a cumprir os seus objetivos:

Utilizador realiza a análise dos diabetes a um sensor usado Dado que o utilizador já colocou a sua amostra de sangue num biossensor, quando inicia a aplicação no *smartphone* ou *tablet* e inicia a digitalização pela câmara e aponta-a para o sensor de forma a ser detetado corretamente, então é apresentado ao utilizador o resultado da análise.

Utilizador realiza a análise dos diabetes a um sensor não usado Dado que o utilizador não colocou a sua amostra de sangue num biossensor, quando inicia a aplicação no dispositivo móvel e inicia a digitalização pela câmara e aponta-a para o sensor de forma a ser detetado corretamente, então é apresentado ao utilizador que o sensor encontra-se vazio.

Utilizador ativa o modo de *Debug* da aplicação Dado que o utilizador tem a aplicação iniciada, quando vai às opções e ativa a opção de depuração, esta opção fica ativada, o que permite que da próxima vez que realiza a análise de um sensor, é apresentado o resultado da sua deteção e posteriormente a cor que foi extraída em conjunto com a informação relativa ao valor de diabetes.

Utilizador visualiza o tutorial pela primeira vez Dado que o utilizador nunca iniciou a aplicação, quando a inicia, é apresentado o tutorial do funcionamento da aplicação, podendo “saltar” se assim o desejar.

Utilizador visualiza mais uma vez o tutorial Dado que o utilizador já visualizou o tutorial, quando se encontra no ecrã principal da aplicação, seleciona a opção para rever o tutorial e este é lhe apresentado.

A partir dos cenários, foi criado um *storyboard* de modo a ilustrar as interações que o utilizador terá com a aplicação que se encontra no apêndice A, assim como um primeiro protótipo, seguido da aplicação final, ambos descritos no capítulo 5.

Além do *storyboard*, existiu a necessidade de um nome para aplicação, sendo para isso escolhido *Mellitus* do nome completo da doença de diabetes, *Diabetes Mellitus*. De seguida, a aplicação necessitava de um logótipo de modo a identificá-la em conjunto com todas as aplicações existentes nos dispositivos móveis, como tal foi criado e usado o ícone que se apresenta na figura 4.1b, consistindo numa máquina regular para medição de diabetes, sendo essa funcionalidade principal do que se desenvolveu. No protótipo da aplicação, apenas a cor de fundo era diferente, sendo esta um tom de cinzento, como se pode verificar na figura 4.1a.



(a) Ícone do protótipo.



(b) Ícone da aplicação final.

4.2 Tecnologias e Arquitetura da Aplicação

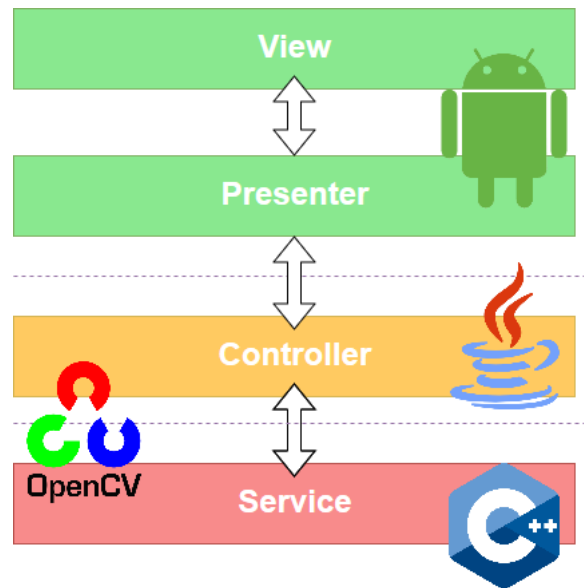


Figura 4.2: Arquitetura das componentes de *Software* da aplicação móvel.

A partir da especificação da secção anterior, decidiu-se desenvolver a aplicação para a plataforma *Android* ¹, sendo esta uma plataforma de sistema operativo de código aberto, e também o sistema móvel com maior quota de mercado existente [22].

Em conjunto com esta plataforma móvel, de modo a realizar as operações necessárias a partir de diferentes imagens, surge a necessidade de uma biblioteca para esse efeito, sendo para isso utilizado o *OpenCV* ². Esta é uma biblioteca também de código aberto que engloba vários algoritmos e operações que podem ser aplicadas a imagens, possuindo também vários algoritmos e funcionalidades de *Machine Learning*. Esta biblioteca, originalmente na linguagem de programação C++ ³ foi integrada na aplicação *Android*, sendo esta desenvolvida com a linguagem de programação *Java* ⁴, comunicando com os algoritmos através da tecnologia *Java Native Interface* (JNI), sendo estes desenvolvidos na mesma linguagem de programação que a do *OpenCV*, neste caso C++.

Na figura 4.2 pode-se verificar a arquitetura da aplicação a partir de um nível mais elevado, dividida em diversas camadas, sendo que as camadas *View* e *Presenter* realizam o tratamento da apresentação da interface e da interação do utilizador com a aplicação, seguida pela camada de *Controller* que recebe as diferentes operações que a aplicação poderá realizar e por fim a camada de *Service* onde é realizada toda a computação das mesmas operações. Esta arquitetura em diferentes camadas apresenta a principal vantagem que qualquer uma das camadas pode ser substituída ou modificada por outras

¹<https://www.android.com>

²<https://opencv.org>

³<https://isocpp.org/>

⁴<https://www.java.com/en/>

instâncias ou implementações diferentes, como o caso de se desejar alocar total ou parcialmente para um servidor a computação de imagem, ou até mesmo integrada em sistemas já existentes, reutilizando o código desenvolvido dos algoritmos. Além disso cada camada pode também ser testada individualmente, durante o processo de desenvolvimento. Todas estas propriedades têm como objetivo a extensibilidade do trabalho aqui desenvolvido em possíveis sistemas médicos novos ou já existentes.

Posta esta arquitetura, de modo a separar as diferentes funcionalidades da aplicação, decidiu-se separá-la em três diferentes componentes, sendo essas:

- Interface Gráfica
- Detecção do biossensor
- Análise Colorimétrica

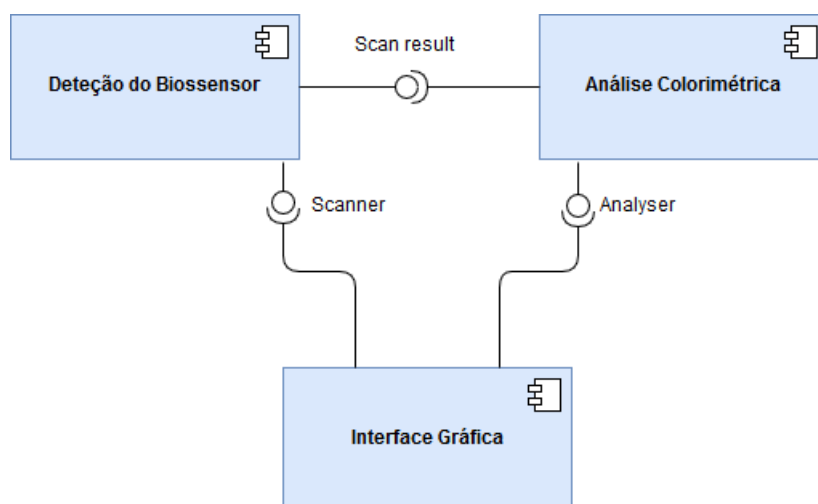


Figura 4.3: Diagrama de componentes da aplicação.

Pela figura 4.3 pode-se verificar um diagrama em *UML* das três componentes, assim como a forma como comunicam entre si. De notar que a partir da componente interface gráfica existe a possibilidade de comunicar de igual forma com a componente de Detecção, e com a componente de Análise Colorimétrica, sendo que esta necessita de comunicar com o resultado da deteção.

IMPLEMENTAÇÃO

Neste capítulo, é apresentada e descrita a implementação do protótipo da aplicação *Mel-litus* desenvolvido na fase de preparação da presente dissertação, seguido pela implementação da sua versão final, sendo esta baseada no protótipo.

De modo a simplificar o trabalho, o desenvolvimento foi dividido nas três componentes descritas no capítulo 4. De notar que os resultados da componente de captura e análise colorimétrica da cor estão dependentes dos resultados da componente de deteção, sendo que existe pelo menos uma região nesse sensor que possui uma dada cor devido a uma reação química aí ocorrida, que se relaciona com a concentração de glucose [6].

5.1 Protótipo

O protótipo desenvolvido teve um foco bastante superior nas componentes relacionadas com processamento de imagem, mais precisamente na deteção do biossensor, de modo a demonstrar o futuro funcionamento da aplicação final, sendo este protótipo apresentado na conferência *Portugal Capital Markets Day* em conjunto com o biossensor da figura 5.1. Como se pode observar na figura, este biossensor possui uma zona com uma mancha escura, onde no seu interior se encontram dois círculos, sendo esses denominados de poços. É nos poços onde se deve colocar a amostra química e onde se encontrará a cor a extrair para a análise, sendo que o poço da esquerda serve para concentrações baixas e o da direita para concentrações altas.

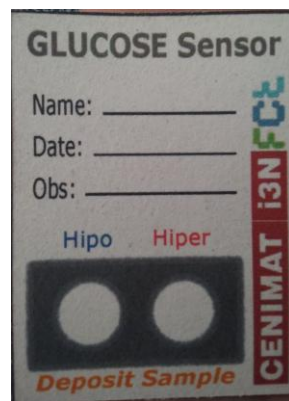


Figura 5.1: Biossensor usado na aplicação de protótipo.

5.1.1 Detecção do Biossensor

A detecção do biossensor consiste, de um modo geral apenas na detecção dos poços numa imagem do sensor. No protótipo, esta componente consiste no passo inicial da captura manual de uma imagem pela câmara do dispositivo móvel, seguido pelo reconhecimento do sensor que poderá lá encontrar-se. Neste passo, a aplicação começa por mostrar ao utilizador a imagem da câmara traseira e, no seu centro, linhas auxiliares de modo a levá-lo a posicionar o sensor corretamente no seu interior.

Após a captura e confirmação da fotografia por parte do utilizador, o algoritmo começa por desenhar na imagem um retângulo na zona correspondente das linhas auxiliares de posicionamento e, com base nas suas medidas, desenha um outro retângulo à volta da zona onde hipoteticamente estarão os respetivos poços, sendo isto feito com base nas medidas do sensor. Assim, de modo a validar a existência de poços nessa zona da imagem, o algoritmo realiza os seguintes passos:

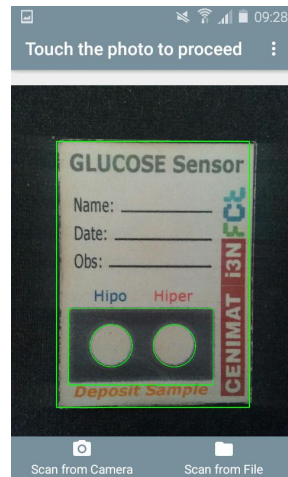
1. Conversão das cores da imagem para tons de cinzento.
2. Binarização da região com base em *threshold* adaptativo.
3. Aplicação de um filtro Gaussiano.
4. Aplicação da *Hough Circle Transform*.

Todos estes passos foram realizados com operações da biblioteca *OpenCV* e parametrizados consoante diversas experiências realizadas com imagens esperadas de detecção com diferentes dispositivos móveis a diferentes iluminações. O resultado do último passo consiste numa coleção de circunferências constituídas pelo seu centro e raio. A partir desta coleção, descartam-se todas as circunferências que se intersejam ou que estejam contidas em outras. Caso não existam só duas circunferências após este passo, é realizado um passo alternativo de binarização da imagem em tons de cinzento com um *threshold* fixo calculado a partir da média dos *pixels* da região e aplica-se mais uma vez a Transformada de *Hough Circles*, procedida pela mesma filtragem.

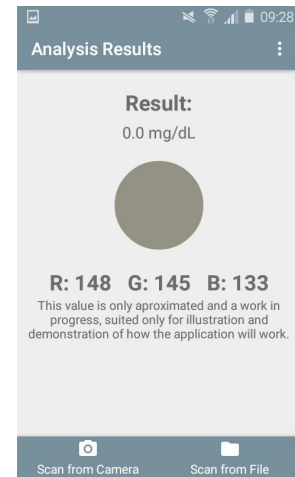
No fim do algoritmo, caso se tenham detetado os dois círculos, estes são ajustados de modo a existir pouca probabilidade de intersetarem a mancha do sensor e também são desenhados na imagem, apresentando no fim o resultado da deteção ao utilizador caso tenha tido sucesso e a aplicação se encontre em modo de depuração, guardando também as regiões da imagem onde se encontram. Um exemplo do resultado da deteção do sensor pode ser observado nas figuras 5.2b e 5.2d.



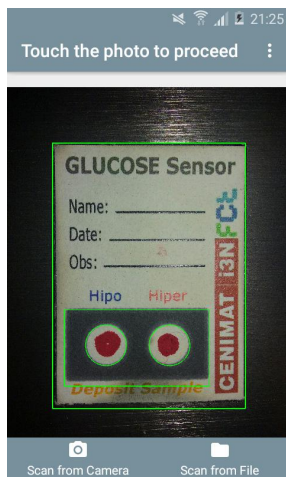
(a) Confirmação da fotografia tirada pelo utilizador.



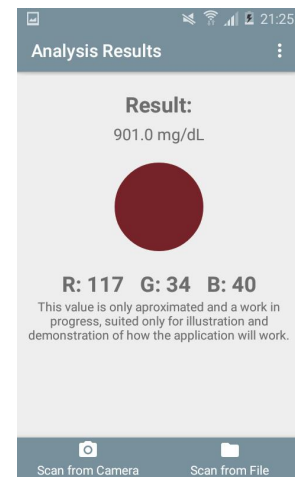
(b) Resultado da deteção do sensor e dos respetivos poços.



(c) Resultado da análise colorimétrica.



(d) Sensor com poço preenchido com tinta vermelha.



(e) Resultado da análise colorimétrica de poço vermelho.

Figura 5.2: Aplicação desenvolvida e os resultados de duas experiências.

5.1.2 Análise Colorimétrica

Após a detecção das regiões dos poços do biossensor, é necessário extrair a cor que lá se encontra. A figura 5.3 mostra uma imagem com poços preenchidos com uma cor vermelha, sendo essa um exemplo da cor a extrair que se encontrará no biossensor. A técnica de extração de cor implementada consiste nos momentos de cor da região dos poços, mais precisamente a média dos *pixels* no formato de cor *RGB*, mas com as regiões dos poços reduzidas a metade do raio dos círculos detetados. Isto de modo a garantir que os *pixels* da mancha negra não influenciem a cor extraída caso os círculos obtidos intersem essa zona.

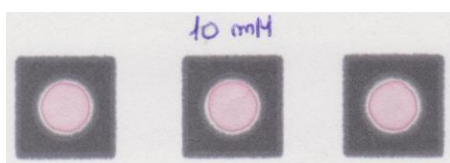


Figura 5.3: Poços preenchidos com cor representativa de concentração de 10 mM (mili-molares) de glucose.

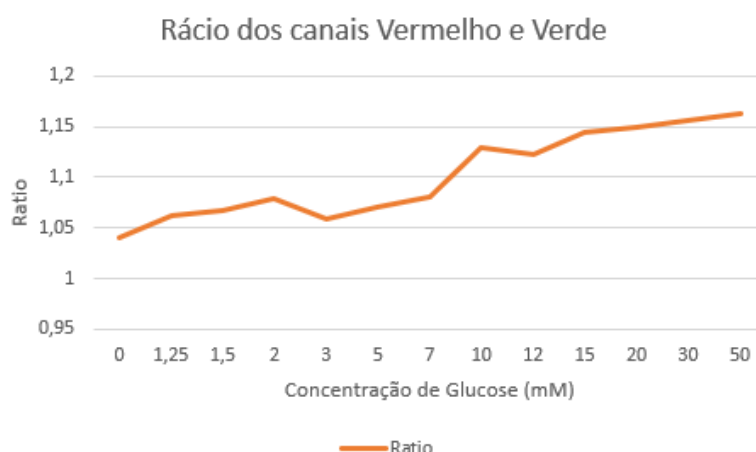


Figura 5.4: Rácio entre o canal Vermelho e o canal Verde da média dos momentos de cor registados para cada concentração de glucose.

No passo seguinte da análise colorimétrica, com os valores obtidos é calculado o rácio entre o canal R e o canal G da cor extraída de cada círculo, considerando o valor máximo entre ambos os rácios como o poço devidamente preenchido, sendo este descritor usado no trabalho realizado anterior [6]. Por fim, com base numa curva de calibração calculada no gráfico da figura 5.4 é determinada a distância mais curta entre os diferentes rácios de modo a mapear-se a concentração de glucose mais próxima, apresentando no final ao utilizador essa concentração convertida para mg/dL, assim como a cor extraída em *RGB* caso a aplicação se encontre em modo de depuração. O resultado da análise pode ser observado na figura 5.2c. De notar que a cor apresentada é influenciada pela iluminação e pela câmara, apresentando uma cor ligeiramente diferente entre dispositivos quando

capturado nas mesmas condições, o que é necessário um passo intermédio de calibração de cor de modo a garantir a cor real.

É também possível observar o resultado de outra experiência que demonstra o reconhecimento da cor extraída pelo resultado das figuras 5.2d e 5.2e. De notar que como indica o ecrã de apresentação da concentração, o valor de concentração apresentado é apenas aproximado.

5.1.3 Interface Gráfica

No protótipo, a interface gráfica foi desenvolvida com base em diversos *guidelines* do *Material Design*, usando as bibliotecas nativas do *Android* para esse efeito. Começando no ecrã inicial, servindo para operações iniciais necessárias, em que é apresentando ao utilizador o logótipo da aplicação durante alguns segundos, sendo depois passado para o ecrã principal. Neste ecrã encontra-se uma barra de menu com duas opções: digitalizar o sensor pela câmara ou digitalizar a partir de um ficheiro, sendo que a segunda opção foi descartada na aplicação final. Além da opção de digitalizar o sensor, existe também uma opção para aceder às definições, de modo a poder-se ativar ou desativar o modo de depuração. Neste ecrã principal também é possível adicionar qualquer informação ou funcionalidade que sejam relevantes para a extensão da aplicação, sendo que nesta fase apenas é apresentado um pequeno texto de contextualização.

Ao seleccionar digitalizar o sensor, o utilizador tem a possibilidade de tirar a fotografia, apresentando a interface descrita na secção 5.1.1 e, a partir de um botão, captura a fotografia, podendo depois confirmá-la ou descartá-la de seguida. Após a imagem ser confirmada, é apresentado ao utilizador uma caixa de diálogo a informar que a deteção está a ser realizada. Caso tenha sucesso, e a aplicação se encontre em modo de depuração, é apresentado o resultado da deteção, com o sensor assinalado e os poços igualmente assinalados na imagem capturada. Caso o sensor não tenha sido detetado, irá aparecer uma mensagem de diálogo como informação, voltando ao ecrã principal. Por fim, caso a aplicação se encontre em modo de depuração e o utilizador tocar na imagem de sucesso da deteção do sensor ou não esteja em depuração e a deteção tenha tido sucesso, é passado para o procedimento de análise colorimétrica, onde é apresentado um diálogo a indicar que está a ser realizada. No fim, é apresentado ao utilizador o resultado da análise, apresentando em conjunto a cor que foi extraída, no modelo *RGB* em conjunto com um círculo preenchido com o valor de cor usado na análise, caso se encontre em modo de depuração.

O *design* da aplicação foi escolhido de modo a fornecer uma interface heterogénea em toda a aplicação, sendo as cores respetivas dos mesmos *guidelines* do *material*. Alguns ecrãs da aplicação podem-se observar na figura 5.2.

5.2 Aplicação Final

A aplicação final foi desenvolvida com base no protótipo, sendo este melhorado e devidamente testado. Nesta aplicação, foi decidido usar-se um biossensor diferente do usado no protótipo, como mostra a figura 5.5. Este sensor, apresenta a vantagem de que apenas um poço servirá para realizar a análise colorimétrica, em conjunto com quatro zonas de calibração de cor, sendo esses círculos o que é necessário detetar.

De seguida segue-se a descrição da implementação de cada uma das componentes.



Figura 5.5: Novo sensor usado para a aplicação final.

5.2.1 Detecção do Biossensor

A deteção do biossensor é realizada igualmente com base nas linhas apresentadas no mostrador da câmara da aplicação, sendo que neste caso, o utilizador deve posicionar o sensor no interior das linhas e a aplicação procura nessa localização da imagem periodicamente. No entanto, esta procura periódica é apenas efetuada caso a imagem se encontre focada. Para isso, foi implementado um algoritmo que verifica se uma dada imagem se encontra desfocada, isto tendo como base 3 casos:

- Uma imagem focada
- uma imagem desfocada devido ao movimento do *smartphone* por parte do utilizador
- uma imagem com o sensor desfocado devido ao foco da câmara

Assim, este algoritmo é constituído pelos seguintes passos:

1. Conversão das cores da imagem para tons de cinzento a partir do *OpenCV*.
2. Aplicação de um filtro de convolução de deteção de contornos chamado *Laplace* na imagem.
3. Cálculo da variância dos *pixels* da imagem resultante do filtro.

Com base neste valor de variância e num *threshold*, considera-se que uma imagem está desfocada se e só se o valor da variância for inferior ao *threshold*. Este valor foi determinado com base em experiências de várias imagens que a aplicação irá receber, sendo estas com o sensor em questão devidamente focado.



(a)

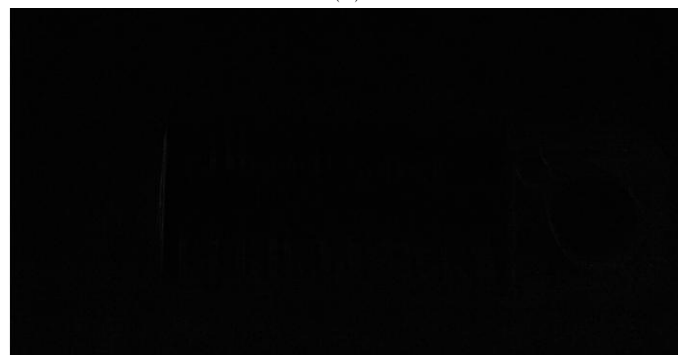


(b)

Figura 5.6: Imagem normal e o resultado do filtro de laplace, apresentando variância de 22.8.



(a)



(b)

Figura 5.7: Imagem desfocada devido ao movimento da câmara e o resultado do filtro de laplace, apresentando variância de 13.19.



(a)



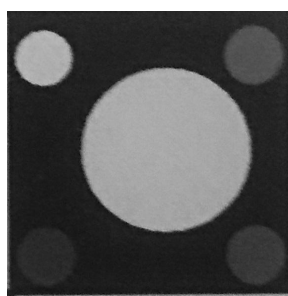
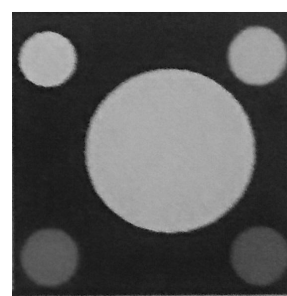
(b)

Figura 5.8: Imagem desfocada e o resultado do filtro de laplace, apresentando variância de 11.53.

Após o passo anterior, caso a imagem encontre-se focada, realiza-se a detecção do sensor. Na detecção, após o cálculo da zona do poço com base nas medidas do sensor, são aplicadas as seguintes transformações na imagem:

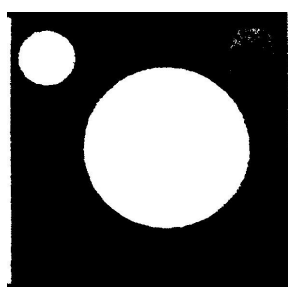
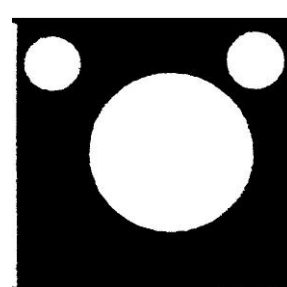
1. Conversão das cores da imagem para tons de cinzento.
2. Aumento do brilho da imagem.
3. Binarização com base em *threshold* adaptativo.
4. Aplicação de um filtro Gaussiano.
5. Aplicação da *Hough Circle Transform*.

A conversão das cores da imagem para tons de cinzento é realizada com base num processo diferente do que está implementado originalmente na biblioteca *OpenCV*, isto porque as zonas de calibração de cor muitas vezes não apresentam contraste suficiente para serem distinguidas na binarização. Por isso, definiu-se o valor cinzento de um dado *pixel* no formato original de cor *RGB* como o valor máximo entre as suas 3 componentes. Pode-se observar na figura 5.9 a diferença de resultados entre as duas conversões de cinzento, notando que as zonas de calibração de cor possuem tons de cinzento mais claros, o que garante uma melhor distinção no passo de binarização.

(a) *OpenCV*.

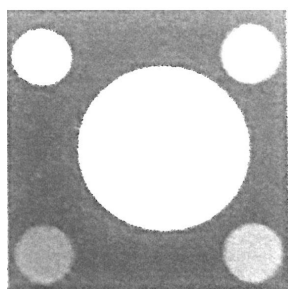
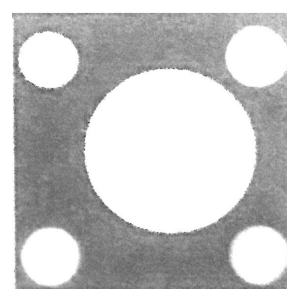
(b) Implementação.

Figura 5.9: Comparação da imagem da zona do poço do biossensor em cinzento, de notar as cores de calibração na implementação.

(a) *OpenCV*.

(b) Implementação.

Figura 5.10: Comparação da imagem da zona do poço do biossensor binarizada, após a conversão para cinzento original e da implementada, sem aumento do brilho.

(a) *OpenCV*.

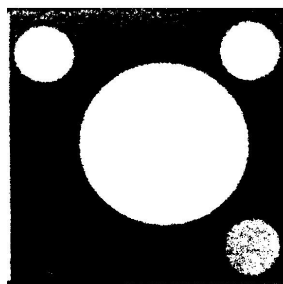
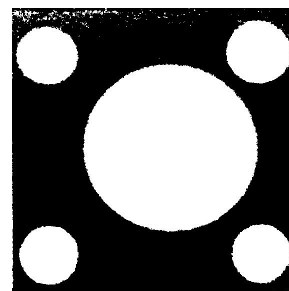
(b) Implementação.

Figura 5.11: Comparação da imagem da zona do poço do biossensor em cinzento após a conversão para cinzento original e da implementada a operação de aumento do brilho.

O passo seguinte de aumento do brilho da imagem é também um passo bastante importante para a deteção de modo a garantir que todas as zonas de calibração de cor possam ser detetadas corretamente. Na aplicação, este aumento do brilho é feito pela seguinte fórmula:

$$P' = P \times r \quad (5.1)$$

Sendo P um dado *pixel* da imagem, em tom de cinzento com valor no intervalo $[0, 255]$, P' o *pixel* final, também no mesmo intervalo e r a razão que se deseja aumentar o valor inicial do pixel, sendo que no caso da aplicação, decidiu-se $r \in \{1, 2, 3, 4, 5\}$. Este valor de

(a) *OpenCV*.

(b) Implementação.

Figura 5.12: Comparação da imagem da zona do poço do biossensor binarizada, após a conversão para cinzento original e da implementada, com aumento do brilho.

r vai variando a cada nova imagem da câmara a que se tenta detetar o sensor, de modo a reduzir o custo de computação para cada imagem. Na figura 5.10 pode-se verificar a diferença dos resultados da binarização da imagem entre as diferentes conversões para tons de cinzento sem aumento do brilho. Na figura 5.11 pode-se verificar o resultado do aumento do brilho na imagem com $r = 4$ em ambas as conversões para cinzento, e na figura 5.12 pode-se verificar o resultado da binarização das imagens da figura anterior.

De seguida, o resultado da aplicação da Transformada *Hough Circles*, ou seja uma coleção de círculos detetados na imagem é filtrado de modo a apenas existirem 5 círculos detetados, removendo quaisquer círculos que intersetem com outros, sendo posteriormente validado com base nessa informação. Verifica-se também se se encontram completamente na zona do poço, apresentando no por fim ao utilizador o resultado caso este tenha sucesso e a aplicação se encontre em modo de depuração. A figura 5.13 mostra a interface gráfica do ecrã de deteção do sensor. No apêndice C encontram-se as imagens resultantes das operações de processamento pelo qual esta componente é constituída, em conjunto com vários testes realizados a diferentes iluminações.

5.2.2 Análise Colorimétrica

Nesta componente, a extração da cor é realizada do mesmo modo que no protótipo, sendo isso feito pelos momentos de cor *RGB* da região do poço previamente detetada na imagem, sendo isto no entanto realizado a 80% do raio. Além do poço onde se encontra a amostra, também são extraídos do mesmo modo os momentos de cor relativos às zonas de calibração, sendo os momentos de cor correspondentes à média dos *pixels*. A extração da cor das zonas de referência deve-se ao passo extra de calibração de cor, sendo descrito a seguir.

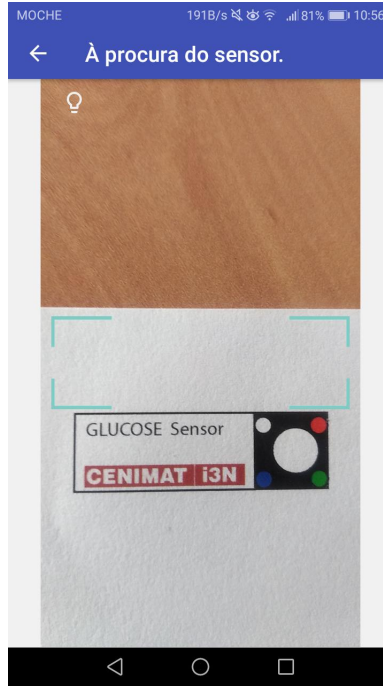


Figura 5.13: Ecrã de deteção do sensor.

5.2.2.1 Calibração de Cor

A calibração de cor é realizada com base nas 4 cores de referência localizadas à volta do poço, sendo essas cores em RGB : Branco, Vermelho, Verde e Azul. Este passo consiste na determinação de uma matriz de calibração de cor M de tamanho 3×3 a partir da resolução de um sistema de equações da forma [4]:

$$A \times M = B \quad (5.2)$$

Em que se deve minimizar:

$$|A \times M| - |B| \quad (5.3)$$

Sendo que:

$$A = \begin{bmatrix} R_b & G_b & B_b \\ R_g & G_g & B_g \\ R_r & G_r & B_r \\ R_w & G_w & B_w \end{bmatrix} \quad B = \begin{bmatrix} R'_b & G'_b & B'_b \\ R'_g & G'_g & B'_g \\ R'_r & G'_r & B'_r \\ R'_w & G'_w & B'_w \end{bmatrix} \quad M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (5.4)$$

Sendo (R_i, G_i, B_i) as cores extraídas da imagem e (R'_i, G'_i, B'_i) as cores de referência já pré-determinadas com $i \in \{r, g, b, w, p\}$, sendo r relativo à zona de calibração do vermelho, g da cor verde, b da cor azul, w da cor branca e p da cor do poço, assim, dada a cor do poço no formato RGB (R_p, G_p, B_p) , a cor calibrada final (R_p^f, G_p^f, B_p^f) é dada pela fórmula:

$$\begin{bmatrix} R_p^f & G_p^f & B_p^f \end{bmatrix} = \begin{bmatrix} R_p & G_p & B_p \end{bmatrix} \times M \quad (5.5)$$

Como cores de referência em *RGB*:

Branco: (229, 228, 230)

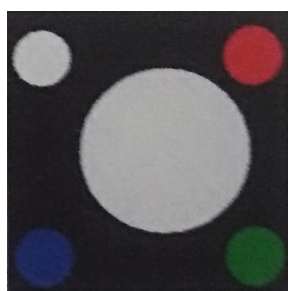
Vermelho: (71, 93, 214)

Verde: (52, 167, 24)

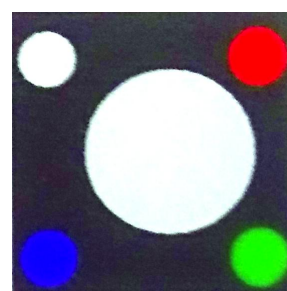
Azul: (171, 85, 60)

Estas cores foram escolhidas com base nos momentos de cor de cada zona de referência extraídos de imagens digitalizadas de biossensores, aproximando assim a cor da imagem da mesma cor das imagens usadas no passo da análise colorimétrica.

Um exemplo do resultado desta transformação da imagem pode ser visualizado na figura 5.14. Esta operação é apenas aplicada à região na figura de modo a reduzir o tempo de processamento deste algoritmo.



(a) Antes de aplicar a transformação.



(b) Depois de aplicada a transformação.

Figura 5.14: Diferença da calibração de cor antes e depois de aplicada a transformação em cada pixel na região do poço, tirado com *Samsung Galaxy S8*.

5.2.2.2 Quantificação Colorimétrica

Neste passo da análise colorimétrica, devido aos resultados pouco precisos a partir da curva de calibração calculada no protótipo, decidiu-se usar uma técnica mais sofisticada através de um algoritmo de *Machine Learning* de classificação, sendo isto um problema de multi-classe. Para tal, foi necessário um conjunto de dados de modo a treinar este algoritmo e criar o seu devido modelo. Posto isto, o *CENIMAT* forneceu várias imagens de várias zonas de cores correspondentes da reação química do biossensor para uma dada concentração de glucose, sendo estas concentrações as correspondentes da tabela 5.1. A figura 5.15 apresenta uma destas imagens usadas para extrair a cor representativa, neste caso de 5 mM, tendo sido extraída a partir de um pequeno programa capaz de detetar os poços através da transformada de *Hough Circles*, extraíndo de seguida a cor a partir dos primeiros momentos de cor *RGB*, criando assim o conjunto de dados que foi usado para criar os modelos.

Devido aos testes e avaliação da eficiência a reconhecer as diferentes concentrações que se podem encontrar na secção 6.2.3, decidiu-se criar dois modelos usados para duas situações: caso o utilizador esteja a medir os diabetes em jejum, convertendo os dados de *RGB* para *HLS* e outra caso o utilizador meça os diabetes 2 horas após a refeição, permanecendo em *RGB*. Para isto, agruparam-se as diferentes concentrações de glucose em intervalos aproximados dos intervalos da tabela 5.2, criando assim os intervalos da tabela 5.3 definindo o seu significado como diferentes classes.

Os dados, antes de serem usados como treino ou como valores para serem previstos pelos modelos, são pré-processados pelos seguintes passos:

1. Normalizar os dados.
2. Aplicação da análise de componentes principais (*PCA*) nos dados.

A normalização dos dados é realizada com base na fórmula:

$$x' = \frac{x - \bar{x}}{\sigma_x^2} \quad (5.6)$$

Sendo x' a observação final, \bar{x} o valor médio de todas as observações e σ_x^2 a variância de todas as observações.

A operação de *PCA* tem a principal vantagem de ser possível facilmente interpretar graficamente o conjunto de dados em 2D com as suas devidas etiquetas, podendo visualizar as fronteiras dos dados, como se pode verificar pelas figuras 5.16, 5.18 e 5.17 em que se reduziu a dimensionalidade de 3 para 2, sendo que pelo primeiro gráfico decidiu-se agrupar as diferentes concentrações de acordo com os valores de referência de diabetes, originando assim os outros dois gráficos.

Concentração (mM)	Valor Diabetes (mg/dL)
0	0
1.25	22.5
1.5	27
2	36
5	90.1
7	126.1
10	180.2
12.5	216.2
15	270.3
20	360.4

Tabela 5.1: Valores de referência para diferentes concentrações de glucose.

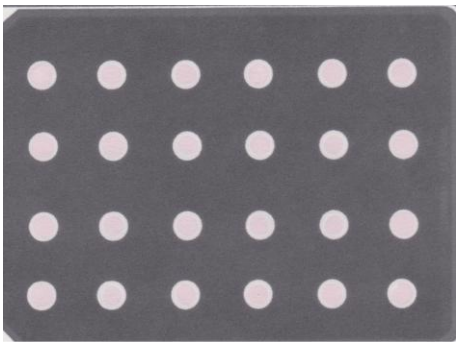


Figura 5.15: Coleção de diferentes poços para a concentração de 5 mM.

	Hipoglicemia	Normal	Pré-diabetes	Diabetes
Jejum	0-36 mg/dl	70-100 mg/dl	90-126 mg/dl	>126 mg/dl
2 horas após refeição	0-36 mg/dl	70-140 mg/dl	90-180 mg/dl	>216 mg/dl

Tabela 5.2: Valores de referência de diabetes [3].

	Hipoglicemia	Normal ou Pré-diabetes	Diabetes
Jejum	0-70 mg/dl	70-126 mg/dl	>126 mg/dl
2 horas após refeição	0-70 mg/dl	70-200 mg/dl	>200 mg/dl

Tabela 5.3: Intervalos de valores usados nos dois modelos e o seu significado.

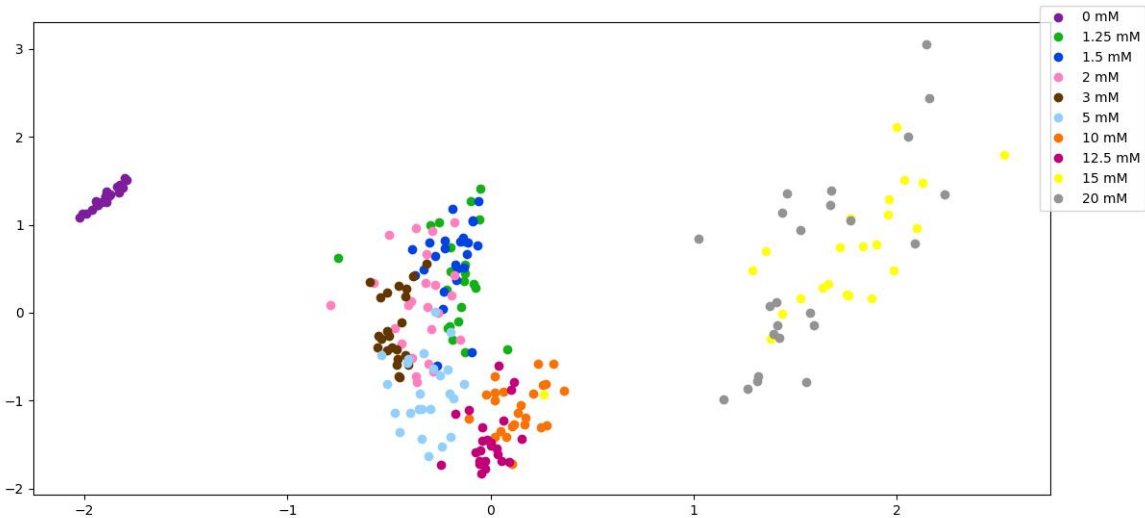


Figura 5.16: Projeção *PCA* do conjunto de dados com o modelo *RGB* do biossensor.

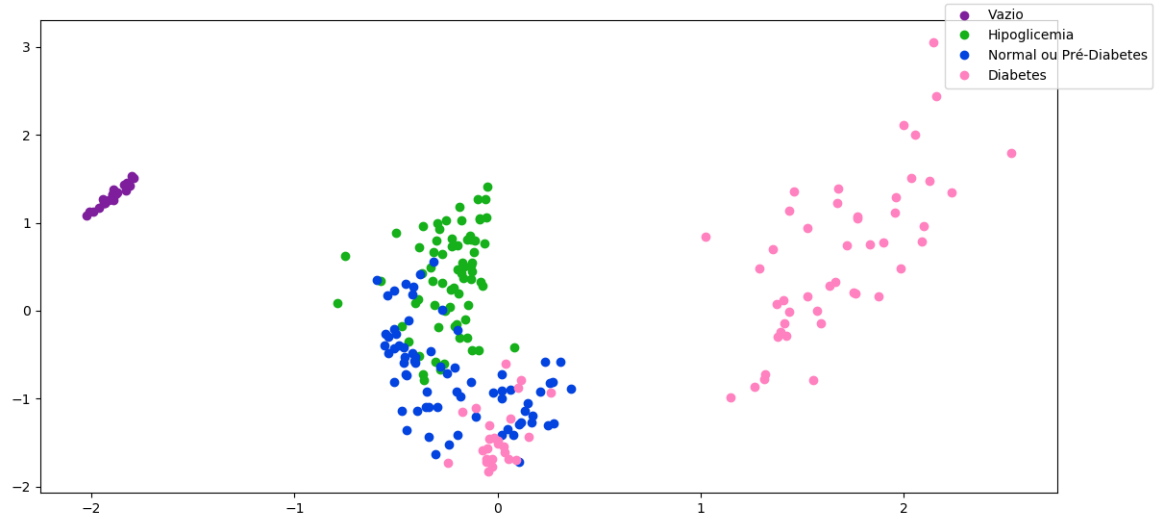


Figura 5.17: Projeção PCA do conjunto de dados com o modelo RGB do biossensor agrupados com base na tabela 5.3, 2 horas após refeição.

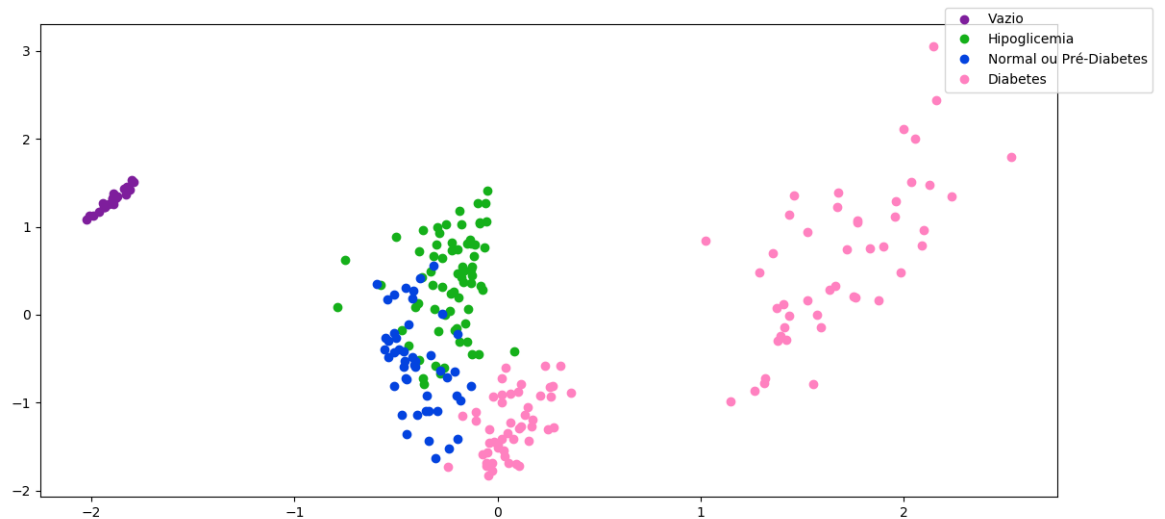


Figura 5.18: Projeção PCA do conjunto de dados com o modelo RGB do biossensor agrupados com base na tabela 5.3, em estado de jejum.

5.2.3 Interface Gráfica

Na aplicação final, a interface gráfica manteve-se praticamente idêntica, tendo sido adicionado um *tutorial* caso o utilizador inicie a aplicação pela primeira vez após aparecer o ecrã com o logótipo. Este tutorial explica como se deve proceder de modo a digitalizar um sensor e verificar-se os resultados da análise, sendo que pode ser visualizado quantas vezes for necessário devido à existência dessa opção no ecrã principal.

No ecrã de digitalização, é apresentado à mesma ao utilizador a câmara traseira, assim como uma zona em que se deve posicionar o respetivo sensor de modo a este ser detetado. Esta funcionalidade em si tem como base na metáfora de digitalização de um *QR Code* ¹, sendo que várias aplicações para esse efeito funcionam de forma idêntica.

Após a digitalização do sensor, caso a aplicação se encontre em modo de depuração, é possível ver os resultados das componentes de deteção e de análise. Para tal, é apresentado ao utilizador o resultado da deteção do sensor aquando do seu sucesso, estando na imagem efetivamente o sensor desenhado, devendo avançar aquando do utilizador tocar na imagem. Ao avançar, a aplicação pergunta ao utilizador se este encontra-se em estado de jejum ou não, assumindo que o sensor foi preparado com a sua respetiva amostra de sangue.

Além destas modificações, a cor principal da interface foi modificada de um tom de cinzento para um tom de azul de modo a diferenciar o protótipo da versão final, assim como fornecer uma cor mais associada a sistemas médicos. No apêndice B encontra-se várias imagens da interface gráfica da aplicação.

¹<http://www.qrcode.com/en/index.html>

TESTES E AVALIAÇÃO

Neste capítulo são descritos os testes realizados no protótipo e na aplicação final de forma a garantir da melhor forma a ausência de erros no *software* desenvolvido, assim como uma maior eficiência nos algoritmos de todas as componentes, apresentando os resultados obtidos e as métricas relevantes de desempenho. Concluindo este capítulo são também apresentados os resultados dos testes de usabilidade da aplicação.

6.1 Protótipo

Na versão protótipo da aplicação, as componentes de detecção do sensor e de análise colorimétrica foram pouco testadas, realçando no entanto os testes da detecção, sendo esta a componente cuja parte da implementação permaneceu como base para a versão final.

6.1.1 Detecção do Biossensor

Esta componente foi a mais testada no protótipo, sendo isto realizado em ambientes de diferentes tipos de iluminação, como luz solar, luz de lâmpadas incandescentes e luz de lâmpadas fluorescentes, usando os *smartphones* *Samsung Galaxy S8* e *Samsung Galaxy Core Prime*. No entanto estes testes foram apenas realizados de modo a garantir que nestes casos em específico, funcionará corretamente, sendo a componente apenas testada, avaliada e corrigida de forma mais completa na aplicação final. No entanto, a partir destes testes surgiu a necessidade da extração da cor da zona dos poços num raio inferior à totalidade do raio dos círculos obtidos, de modo a garantir que os *pixels* ao redor dos poços não afetem os valores da cor extraída pelos primeiros momentos de cor *RGB*.

6.1.2 Análise Colorimétrica

Nesta componente, a precisão da análise dos diabetes não foi testada de forma tão completa como a componente anterior, sendo que devido à sua implementação numa curva de calibração determinada a partir de algumas cores para cada concentração de glucose, apresentou resultados pouco precisos nas diferentes cores, mostrando também a necessidade de uso de uma técnica mais sofisticada de forma a obter resultados melhores. Além dos resultados pouco precisos, surgiu também a necessidade de um passo de calibração de modo a ser possível extrair a cor o mais próxima possível da cor real do poço, independentemente da iluminação e da câmara.

6.2 Aplicação Final

Na aplicação final, de um modo geral foram realizados diversos testes a diferentes iluminações com diferentes câmaras de dispositivos móveis, isto em conjunto com diversos testes na componente de análise colorimétrica, mais precisamente na geração dos melhores modelos de classificação de modo a obter os melhores resultados possíveis, sendo estes medidos através de algumas métricas usadas para medir o erro que um típico modelo de classificação de *Machine Learning* apresenta.

6.2.1 Detecção do Biossensor

Esta componente foi testada a partir de câmaras de *smartphone* de diferentes qualidades de imagem, em combinação com diferentes tipos de iluminação, sendo esses: luz natural, luz incandescente e luz fluorescente. Sendo as câmaras de: baixa, média e alta qualidade pertencentes aos modelos *Samsung Galaxy Core Prime*, *Huawei Y7* e *Samsung Galaxy S8* respetivamente. Adicionalmente foram usadas algumas imagens do *smartphone Samsung Galaxy Note 3*, um dispositivo com uma câmara de qualidade mais baixa e capturadas a luz natural e luz incandescente. Os resultados de cada um dos passos estão na secção C.2 do respetivo apêndice.

Sendo todas as imagens capturadas a uma distância de aproximadamente 10 cm, pôde-se verificar pelos resultados que foi possível detetar todos os sensores em todas as diferentes iluminações. No entanto existem casos que o círculo detetado do poço intersesta a zona negra ao seu redor, o que pode afetar a extração da cor. Isso é causado pela operação de binarização, que aumenta ligeiramente o raio do poço na imagem, algo que já acontecia na versão de protótipo. Este problema deixa de ser tão relevante ao extrair a cor do poço a um raio menor do que o detetado, tendo sido para tal escolhido a 80% do raio original.

6.2.2 Calibração de Cor

Esta componente, tal como a componente de deteção, foi testada nas mesmas condições descritas com os diferentes tipos de câmaras de *smartphone* em diferentes iluminações. Pode-se verificar pelos testes na secção C.2 do respetivo apêndice que, comparando as duas figuras de cada teste, as cores se vão aproximando das cores reais do sensor, no entanto nem sempre são exatas em diferentes câmaras ou em outros testes com diferentes iluminações, podendo-se verificar em vários casos um tom de cor azulado em imagens com sensores de poços vazios. Este problema poderá ser resolvido com a inclusão de um maior número de zonas de calibração de cor com diferentes cores de referência.

6.2.3 Análise Colorimétrica

Devido à natureza da implementação desta componente, esta foi testada e avaliada ao mesmo tempo que implementada de modo a se decidir quais os modelos mais adequados e mais bem comportados para este problema. Como possíveis algoritmos de classificação, foram usados três dos mais conhecidos e comuns sendo esses *Support Vector Machines*, *K-Nearest Neighbors* e *Random Trees*. Todos estes testes e avaliações foram realizados com base no conjunto de dados obtido e descrito na secção 5.2.2.2 sendo as cardinalidades de cada concentração as que se encontram na tabela 6.1. O conjunto de dados foi convertido do modelo de cor original *RGB* para: o rácio entre o *R* e o *G* e para os modelos de cor *YUV*, *HLS*, *HSV* e *XYZ*. Após a conversão para os diferentes modelos de cor, todos os dados foram de seguida pré-processados de igual modo como se encontra descrito na mesma secção 5.2.2.2 em todos os testes, tendo sido comparado esse procedimento com outro idêntico mas sem o uso do *PCA*.

O procedimento de avaliação inicia-se pela divisão de todos os conjuntos de dados de forma aleatória em dois subconjuntos: um de treino e outro de teste. Com o conjunto de treino, em combinação com a técnica de *Grid Search*, decide-se quais são os melhores parâmetros dos algoritmos, medindo as métricas de *Precision*, *Recall* e *Fscore*, através de *K-Fold Cross-Validation* com $k = 10$, ajustando os parâmetros de modo a maximizar o valor de *Fscore*, com $\beta = 1$, sendo essas métricas as escolhidas conforme o contexto do problema.

Com os parâmetros ajustados, as mesmas métricas foram calculadas com o conjunto de teste, usando de igual modo o *K-Fold Cross-Validation* com $k = 10$ e registadas de modo a determinar aproximadamente como cada modelo se comporta com dados novos. Assim, pelos resultados obtidos e registados nos apêndices D e E, pode-se verificar em primeiro lugar que independentemente do modelo de cor, se obtiveram resultados não muito elevados na classificação para concentrações de glucose, segundo os gráficos correspondentes no apêndice E. Estes gráficos mostram que os modelos necessitaram de se ajustar demasiado aos dados, possuindo pouca capacidade de reconhecer valores de biossensores fora do conjunto de dados com que foram treinados. Devido a isto, e com base na representação gráfica dos dados que se pode observar na figura 5.16 e nos valores de referência da tabela 5.2, foi decidido agrupar os diferentes pontos do conjunto de dados em novas classes de

Concentração (mM)	Número de observações
0	24
1.25	23
1.5	24
2	24
5	23
7	24
10	24
12.5	24
15	23
20	24

Tabela 6.1: Número de observações no conjunto de dados usado para criar os modelos, fazendo no total 237 observações.

Resultado da análise	Número de observações
Vazio	24
Hipoglicemia	71
Normal ou Pré-Diabetes	47
Diabetes	95

Tabela 6.2: Número de observações no conjunto de dados usado para criar os modelos finais para utilizadores em jejum.

classificação. E para tal, foram criados dois modelos diferentes: um para caso o utilizador se encontre em estado de jejum e outro caso se encontre 2 horas após uma refeição, sendo as cardinalidades para cada classe as que se encontram nas tabelas 6.2 e 6.3 respetivamente. Pelos testes com estes “novos” conjunto de dados, determinaram-se dois modelos ideais para os dois casos, usando como modelos de cor o *HLS* e *RGB* respetivamente: para o estado de jejum a partir do algoritmo *Support Vector Machines*, com as métricas na tabela 6.4 e para uma situação de 2 horas após a refeição o algoritmo *K-Nearest Neighbors*, com as métricas que se podem verificar na tabela 6.6, sendo também estes dois modelos parametrizados treinados com todo o conjunto de dados disponível. Por estes modelos, verifica-se que o uso do *PCA* origina melhores resultados de classificação, assim como que permite a fácil visualização das fronteiras pois os dados se encontram a 2 dimensões. Em conjunto com os gráficos das figuras 6.1 e 6.2 pode-se também verificar que ambos estes modelos possuem uma elevada capacidade de reconhecer biossensores que se encontrem vazios e representativos de Hipoglicemia ou de Diabetes, sendo que estes dois últimos resultados são bastante importantes de modo a diagnosticar corretamente a doença.

De notar, no entanto, que devido à baixa cardinalidade do conjunto de dados obtido, a repetição da geração dos modelos pode gerar resultados com métricas que variem ligeiramente, sendo que apenas foi descrita nos respetivos apêndices uma execução desses testes.

Resultado da análise	Número de observações
Vazio	24
Hipoglicemia	71
Normal ou Pré-Diabetes	71
Diabetes	71

Tabela 6.3: Número de observações no conjunto de dados usado para criar os modelos finais para utilizadores que tomaram uma refeição há 2 horas.

Resultado da Análise	Precision (%)	Recall (%)	Fscore (%)
Vazio	100	100	100
Hipoglicemia	85	89	87
Normal ou Pré-Diabetes	92	80	86
Diabetes	97	100	99

Tabela 6.4: Modelo desenvolvido e usado de *Support Vector Machines* para utilizador em estado de jejum usando o modelo de cor *HLS*.

Resultado da Análise	Precision (%)	Recall (%)	Fscore (%)
Sensor Vazio	100	100	100
Hipoglicemia	89	83	86
Normal ou Pré-Diabetes	72	81	76
Diabetes	100	100	100

Tabela 6.5: Modelo desenvolvido de *Support Vector Machines* para utilizador em estado de jejum usando o modelo de cor *HLS* sem *PCA*.

Resultado da Análise	Precision (%)	Recall (%)	Fscore (%)
Vazio	100	100	100
Hipoglicemia	81	89	85
Normal ou Pré-Diabetes	82	78	80
Diabetes	93	89	91

Tabela 6.6: Modelo desenvolvido e usado de *K-Nearest Neighbors* para utilizador 2 horas após refeição usando o modelo de cor *RGB*.

Resultado da Análise	Precision (%)	Recall (%)	Fscore (%)
Sensor Vazio	100	100	100
Hipoglicemia	86	83	85
Normal ou Pré-Diabetes	77	071	074
Diabetes	84	100	91

Tabela 6.7: Modelo desenvolvido de *K-Nearest Neighbors* para utilizador 2 horas após refeição usando o modelo de cor *RGB* sem *PCA*.

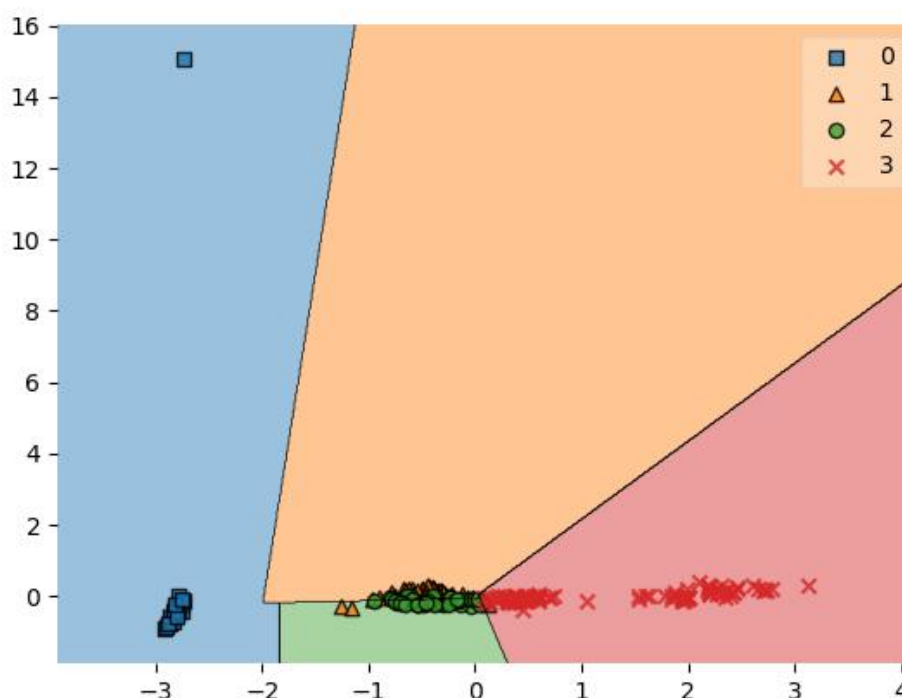


Figura 6.1: Regiões de decisão do modelo para utilizador em estado de jejum.

6.2.4 Testes de Integração das Componentes

De modo a testar a aplicação completa, a combinação de todas as componentes foi testada de modo a avaliar o desempenho do que se desenvolveu, em contexto real. Para tal, foram usados os mesmos sensores que nos testes das componentes anteriores nas mesmas condições de luz e com as mesmas respetivas câmaras de *smartphone*. Por estes testes realizados, pôde-se concluir que além de ter sido possível detetar todos os biossensores a iluminações diferentes nas várias câmaras, como já foi descrito na secção de testes da respetiva componente, nem todos os sensores conseguiram ser analisados corretamente. Isto deve-se principalmente ao passo da calibração de cor, visto que a cor extraída resultante da calibração é diferente da cor real, sendo esta muitas vezes um tom de azul ou verde quando devia ser um tom de vermelho, ou branco no caso de um poço vazio. Como foi descrito também na secção de testes 6.2.2, este problema pode ser resolvido ajustando a calibração ao adicionar mais zonas de referência ao biossensor. Em alternativa pode-se realizar a análise a uma iluminação mais natural que aumentará a eficiência dos resultados do sensor enquanto não existe a possibilidade dessas zonas de calibração. Além destes passos, também de modo a obter-se melhores resultados seria ideal aumentar o conjunto de dados de modo a deixar todas as classes equilibradas em ambos os modelos, aumentando também a sua cardinalidade e consequentemente o desempenho da classificação.

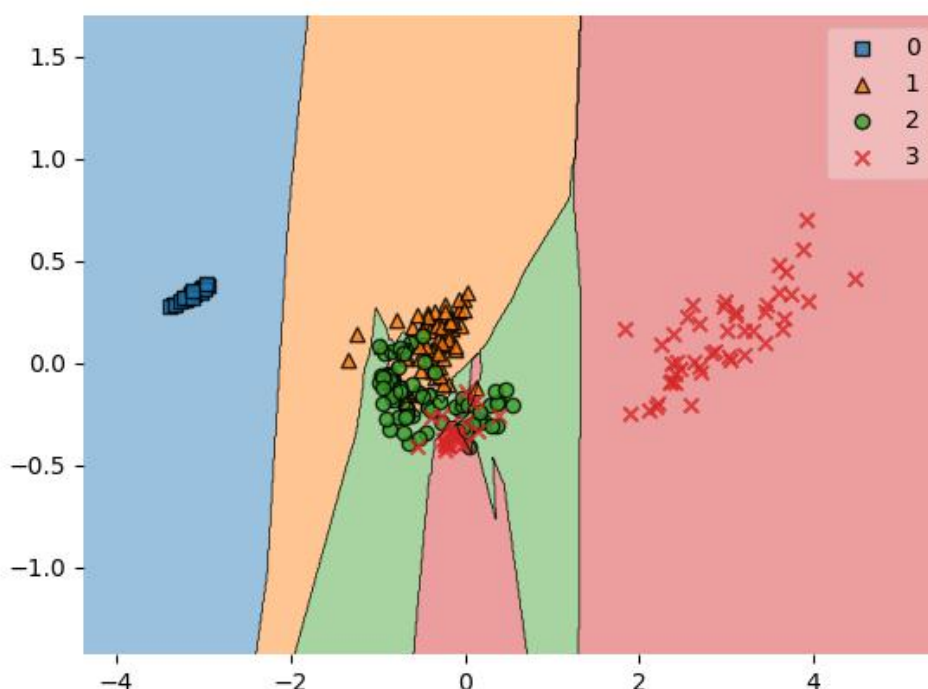


Figura 6.2: Regiões de decisão do modelo para utilizador 2 horas após refeição.

6.2.5 Teste de Utilizadores

Este tipo de teste serviu para avaliar a usabilidade da aplicação, colocando potenciais utilizadores a realizar alguns dos cenários descritos na secção 4.1 e por fim, responderem a um questionário sobre a sua interação com a interface gráfica, obtendo desta forma diferentes opiniões sobre o uso da aplicação. Este questionário foi feito com base em respostas em escalas de quanto o utilizador concorda ou discorda sobre como se sentiu ao utilizar a aplicação e encontra-se disponível em conjunto com as percentagens de respostas no apêndice F.

Este questionário foi realizado a uma população maioritariamente jovem, com algumas pessoas com diabetes ou que tenham conhecimento de familiares ou amigos que sofram dessa doença. Segundo os resultados, os utilizadores que testaram a aplicação consideraram que a deteção do biossensor era de um modo geral bastante fácil, embora tenham notado alguma dificuldade inicial em descobrir como a realizar. Isto deve-se principalmente ao aumento do brilho na imagem, assim como à dificuldade em posicionar o sensor, tendo para isso surgido como proposta colocar umas marcas no sensor correspondentes às linhas de posicionamento que se encontram na câmara de modo a que o utilizador entenda de forma mais clara de que maneira as deve posicionar.

Vários utilizadores consideraram que a informação da análise era útil, realçando no entanto alguns casos que muitos pacientes com diabetes necessitam de controlar a doença

de forma quantitativa, o que na aplicação não é possível com precisão alta devido à capacidade do sensor. Também foi realçado o potencial da facilidade de uso e de portabilidade por comparação a uma máquina de diabetes convencional. Para isso, bastaria possuir um pequeno *kit* de preparação da amostra química, em conjunto com alguns sensores e a aplicação instalada de modo a poder verificar se possuía uma concentração elevada ou baixa. Isto facilitaria o diagnóstico da doença em alguns casos mais "rápidos" ou mesmo como um primeira análise.

No geral, pelas respostas dos utilizadores concluímos que a aplicação foi considerada simples e acessível de usar, embora com um uso bastante específico, não substituindo por completo as máquinas usuais de análise de diabetes, surgindo assim a necessidade de integrar num sistema médico mais complexo ou, em combinação com uma melhoria dos resultados do biossensor, tornar a aplicação a apresentar, em combinação com valores qualitativos, também valores quantitativos.

CONCLUSÕES E TRABALHO FUTURO

Nesta dissertação foi possível realizar e completar todos os objetivos que foram propostos, nomeadamente foi possível desenvolver uma aplicação que realize a análise colorimétrica de concentração de glucose a partir da imagem de um biossensor com resultados de desempenho e de precisão de análise elevados, tendo em conta a característica que este sensor possui de distinguir concentrações elevadas de baixas. No entanto, não foi possível distinguir quantitativamente a concentração com alta eficiência, como os testes o demonstraram. Este importante passo de análise foi possível graças ao desenvolvimento de dois modelos de *Machine Learning* capazes de distinguir o significado das diferentes cores.

Com as funcionalidades desenvolvidas, existe um elevado potencial para serem integradas com sistemas médicos complexos e robustos de modo a ser possível controlar os valores de diabetes dos utentes em contexto clínico, podendo ser implementado também em diferentes plataformas móveis, inclusive com uma arquitetura que funcione com base em servidor. No entanto, de modo a existir um sistema que funcione melhor, é primeiro necessário aumentar o tamanho do conjunto de dados, de modo a obter uma representação estatística mais exata de cada uma das classes, equilibrando e aumentando as suas respectivas cardinalidades. Além disso é também necessário adicionar mais cores de calibração ao sensor de forma a ser possível calcular uma cor mais próxima da cor real. Além das cores de calibração, também melhorias nas cores geradas pelo biossensor com base na concentração de glucose são importantes para o mesmo efeito, sendo assim todos estes pontos propostas para um trabalho futuro de continuação do que aqui foi realizado.

A partir do teste de usabilidade em conjunto com possíveis utilizadores, foi possível concluir que além da aplicação ser simples de usar, apresenta resultados que possam ser relevantes principalmente para análises mais rápidas ou como forma de interpretar mais facilmente os resultados. Apesar disso, como melhoria do sistema, foi sugerida a

possibilidade de obter valores quantitativos na aplicação em conjunto com os valores já usados, mesmo que estejam em forma de intervalos, isto de modo a melhorar o diagnóstico e obter mais informação para utilizadores que necessitam de controlar a doença de um modo mais detalhado.

Outras sugestões de outros participantes no teste envolveram melhorias na interface gráfica ou do sensor, como ao iniciar a aplicação, esta ir diretamente para o ecrã de digitalizar o sensor, ou associar cores a diferentes resultados da análise ou mesmo adicionar linhas de posicionamento num futuro biossensor, levando o utilizador a entender como deve tentar detetar o sensor.

BIBLIOGRAFIA

- [1] W. M. Afifi Ahmed J.;Ashour. "Image Retrieval Based on Content Using Color Feature". Em: *ISRN Computer Graphics* 2012 (2012). DOI: [10.5402/2012/248285](https://doi.org/10.5402/2012/248285). URL: <http://dx.doi.org/10.5402/2012/248285>.
- [2] E. Alpaydin. *Introduction to Machine Learning, Second Edition*. MIT Press, 2010. ISBN: 978-0-262-01243-0.
- [3] Apdp. *Valores de Referência*. URL: <https://www.apdp.pt/diabetes/a-pessoa-com-diabetes/valores-de-referencia#valores-de-referencia>.
- [4] A. Behringer. "Camera Array Calibration with Color Rendition Charts". Tese de mestrado. Berlin: Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät, Institut für Informatik, 2013.
- [5] S. P. de Diabetologia. "Diabetes - Factos e Números o ano de 2015. Relatório anual do observatório nacional da diabetes - Edição 2016". Em: (2016).
- [6] J. Ferrão. "Aplicação da tecnologia de baixo custo Lab-on-Paper a biossensores colorimétricos não enzimáticos para a deteção de glucose". Tese de mestrado. Campus Caparica Portugal: Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa, 2016.
- [7] A. K. Gary Bradski. *Learning OpenCV*. O'Reilly, 2008. ISBN: 978-0-596-51613-0.
- [8] Google. *Material Design*. Accessed 08-08-2018. URL: <https://material.io/design/introduction/#principles>.
- [9] Google. *Material Studies Rally*. Accessed 09-08-2018. URL: <https://material.io/design/material-studies/rally.html>.
- [10] D. C. Hay Burgess, J. Wasserman e C. A. Dahl. "Global health diagnostics". Em: *Nature* 444 (2006), 1 EP -. URL: <http://dx.doi.org/10.1038/nature05440>.
- [11] J. Hu, S. Wang, L. Wang, F. Li, B. Pingguan-Murphy, T. J. Lu e F. Xu. "Advances in paper-based point-of-care diagnostics". Em: *Biosensors and Bioelectronics* 54 (2014), pp. 585 -597. ISSN: 0956-5663. DOI: <https://doi.org/10.1016/j.bios.2013.10.075>. URL: <http://www.sciencedirect.com/science/article/pii/S095656631300777X>.

- [12] J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, J. D. Foley, S. K. Feiner e K. Akeley. *Computer graphics: principles and practice (3rd ed.)* Boston, MA, USA: Addison-Wesley Professional, 2013, p. 1264. ISBN: 0321399528.
- [13] A. Koh, D. Kang, Y. Xue, S. Lee, R. M. Pielak, J. Kim, T. Hwang, S. Min, A. Banks, P. Bastien, M. C. Manco, L. Wang, K. R. Ammann, K.-I. Jang, P. Won, S. Han, R. Ghaffari, U. Paik, M. J. Slepian, G. Balooch, Y. Huang e J. A. Rogers. “A soft, wearable microfluidic device for the capture, storage, and colorimetric sensing of sweat”. Em: *Science Translational Medicine* 8.366 (2016), 366ra165–366ra165. ISSN: 1946-6234. DOI: [10.1126/scitranslmed.aaf2593](https://doi.org/10.1126/scitranslmed.aaf2593). eprint: <http://stm.sciencemag.org/content/8/366/366ra165.full.pdf>. URL: <http://stm.sciencemag.org/content/8/366/366ra165>.
- [14] J. Lewis e J. R. “IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use”. Em: 7 (jan. de 1993), pp. 57–.
- [15] A. Y. Mutlu, V. K[i without dot]l[i without dot]c, G. K. Ozdemir, A. Bayram, N. Horzum e M. E. Solmaz. “Smartphone-based colorimetric detection via machine learning”. Em: *Analyst* 142 (13 2017), pp. 2434–2441. DOI: [10.1039/C7AN00741H](https://doi.org/10.1039/C7AN00741H). URL: <http://dx.doi.org/10.1039/C7AN00741H>.
- [16] J. Nielsen. *Usability Engineering*. Academic Press, 1993. ISBN: 1-12-518406-9.
- [17] OpenCV. *Canny Detector*. Accessed 23-01-2018. URL: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html/.
- [18] W. H. Organization. “Global Reports On Diabetes”. Em: (2016).
- [19] D. Quesada-González e A. Merkoçi. “Mobile phone-based biosensing: An emerging “diagnostic and communication” technology”. Em: *Biosensors and Bioelectronics* 92.Supplement C (2017), pp. 549 –562. ISSN: 0956-5663. DOI: <https://doi.org/10.1016/j.bios.2016.10.062>. URL: <http://www.sciencedirect.com/science/article/pii/S095656631631082X>.
- [20] P. Salembier. *Introduction to MPEG 7: Multimedia Content Description Language*. 1ª ed. 2002. ISBN: 0471486787,9780471486787,9780470852408.
- [21] M. Sokolova e G. Lapalme. “A systematic analysis of performance measures for classification tasks”. Em: *Information Processing & Management* 45.4 (2009), pp. 427 –437. ISSN: 0306-4573. DOI: <https://doi.org/10.1016/j.ipm.2009.03.002>. URL: <http://www.sciencedirect.com/science/article/pii/S0306457309000259>.
- [22] Statista. *Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2017*. Accessed 11-01-2018. URL: <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>.

-
- [23] J. Suykens e J. Vandewalle. “Least Squares Support Vector Machine Classifiers”. Em: *Neural Processing Letters* 9.3 (1999), pp. 293–300. ISSN: 1573-773X. DOI: [10.1023/A:1018628609742](https://doi.org/10.1023/A:1018628609742). URL: <https://doi.org/10.1023/A:1018628609742>.
- [24] R. Szeliski. *Computer Vision: Algorithms and Applications*. 1st. New York, NY, USA: Springer-Verlag New York, Inc., 2010. ISBN: 1848829345, 9781848829343.
- [25] X-Rite. *ColorChecker Passport*. accessed 10-03-2018. URL: <https://xritephoto.com/colorchecker-passport-photo>.
- [26] A. K. Yetisen, J. Martinez-Hurtado, A. Garcia-Melendrez, F. da Cruz Vasconcellos e C. R. Lowe. “A smartphone algorithm with inter-phone repeatability for the analysis of colorimetric tests”. Em: *Sensors and Actuators B: Chemical* 196.Supplement C (2014), pp. 156 –160. ISSN: 0925-4005. DOI: <https://doi.org/10.1016/j.snb.2014.01.077>. URL: <http://www.sciencedirect.com/science/article/pii/S092540051400094X>.
- [27] A. K. Yetisen, M. S. Akram e C. R. Lowe. “Paper-based microfluidic point-of-care diagnostic devices”. Em: *Lab Chip* 13 (12 2013), pp. 2210–2251. DOI: [10.1039/C3LC50169H](https://doi.org/10.1039/C3LC50169H). URL: <http://dx.doi.org/10.1039/C3LC50169H>.

A P Ê N D I C E



Storyboard da Aplicação

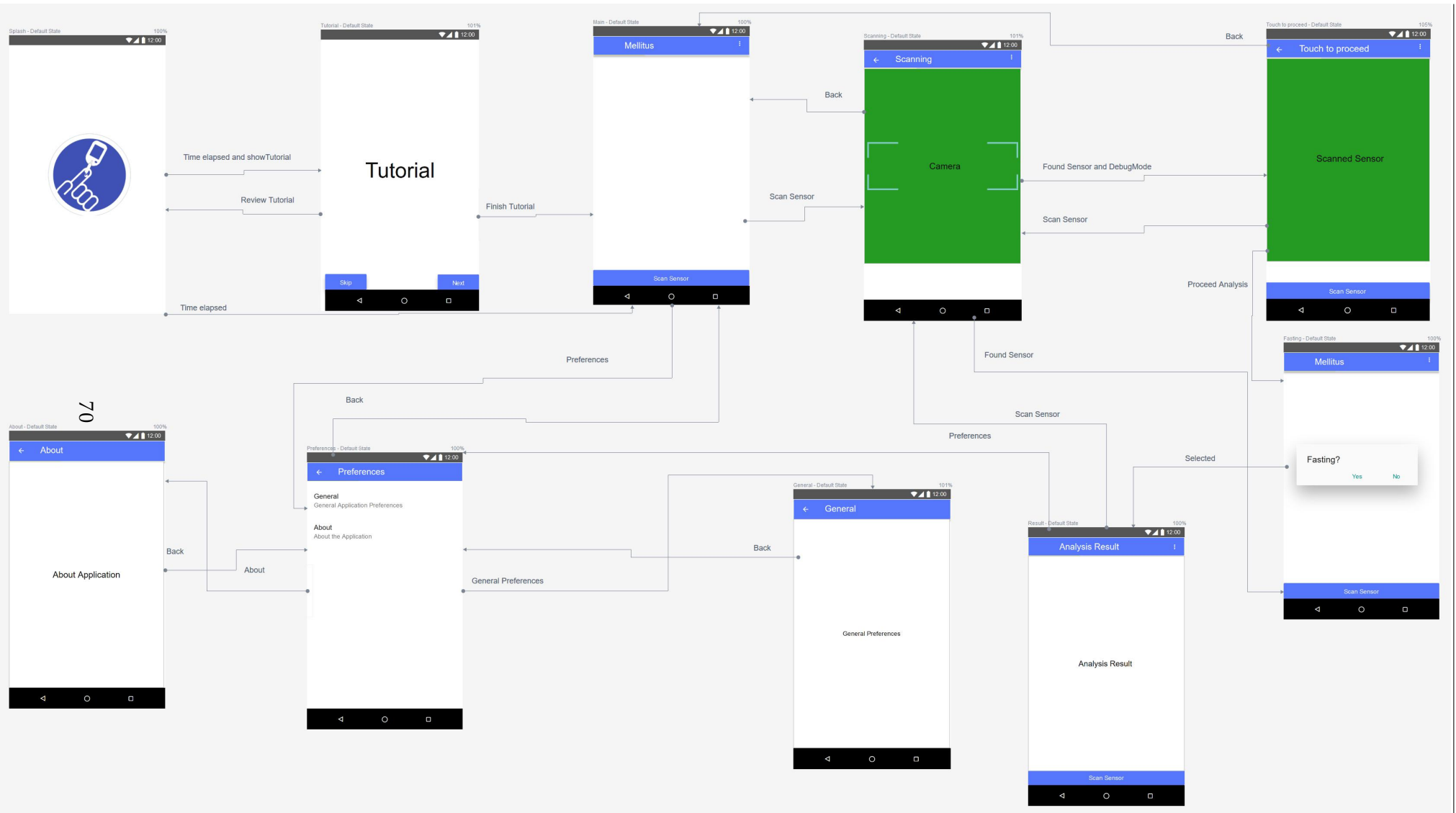


Figura A.1

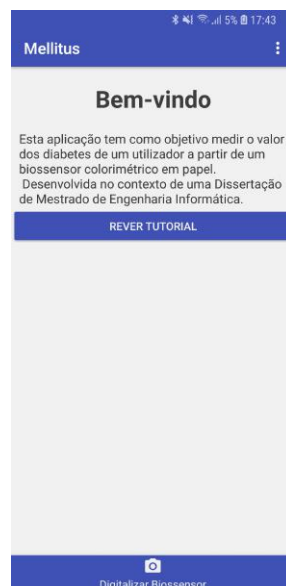
INTERFACE GRÁFICA DA *Mellitus*

Neste apêndice encontram-se *screenshots* da aplicação *Mellitus* desenvolvida nesta dissertação. Estas imagens foram tiraas no *smartphone Samsung Galaxy S8*.

B.1 Ecrãs Principais



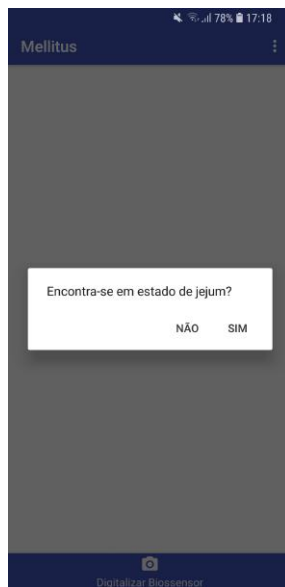
(a) Ecrã inicial de processamento.



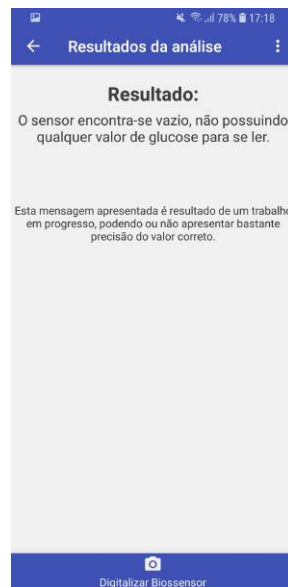
(b) Ecrã inicial.



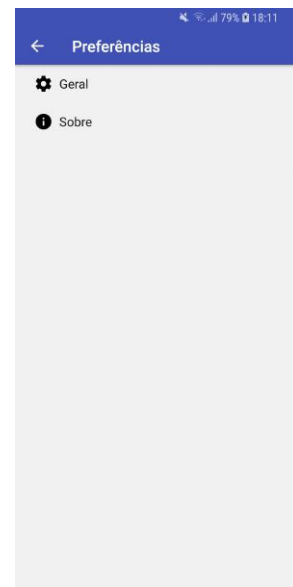
(c) Ecrã de digitalização.



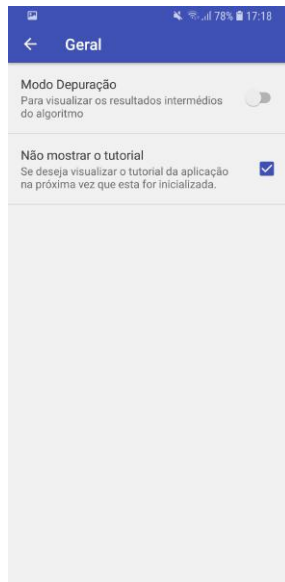
(a) Ecrã de questionar o utilizador.



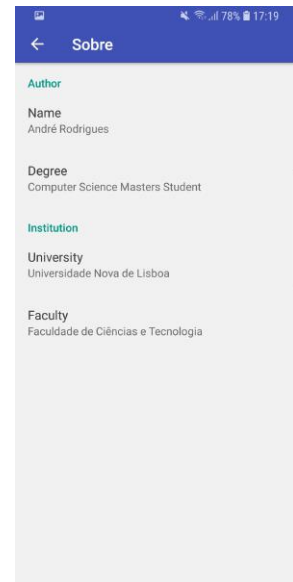
(b) Ecrã de resultados.



(c) Ecrã de preferências.



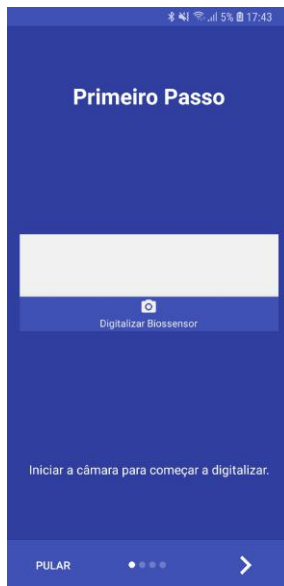
(d) Ecrã de preferências gerais.



(e) Ecrã sobre o autor da aplicação.

Figura B.1: Interfaces gráficas principais da aplicação.

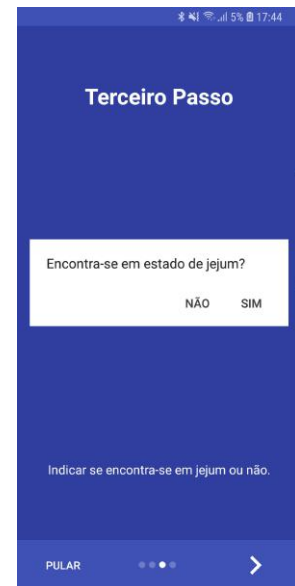
B.2 Ecrãs do *Tutorial*



(a) Ecrã 1 do *tutorial*.



(b) Ecrã 2 do *tutorial*.



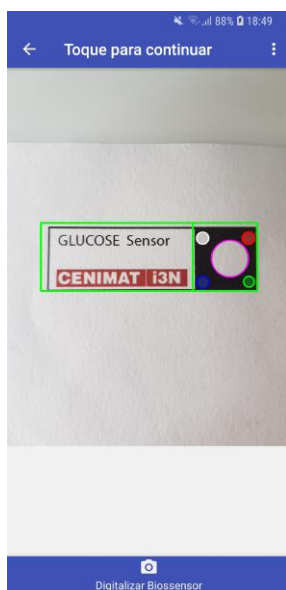
(c) Ecrã 3 do *tutorial*.



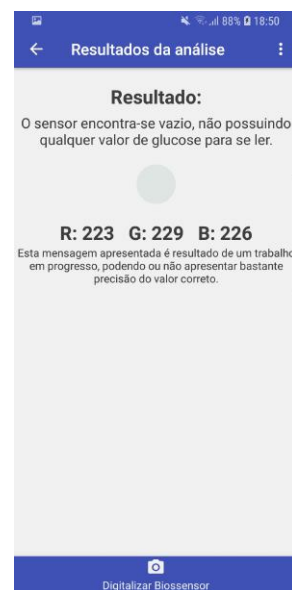
(a) Ecrã 4 do *tutorial*.

Figura B.2: Interfaces gráficas do *tutorial*

B.3 Ecrãs de Depuração



(a) Ecrã de resultado da deteção em depuração.



(b) Ecrã de resultado em depuração.

Figura B.3: Interfaces gráficas adicionais de depuração.



RESULTADOS DOS TESTES DE DETECÇÃO DO BIOSSENSOR E CALIBRAÇÃO DE COR

Neste apêndice são apresentados os resultados dos testes realizados na componente de detecção do biossensor, os da calibração de cor e os resultados das análises nas tabelas C.1 e C.2. Estes resultados estão divididos em tipos diferentes de iluminação a que foram submetidos. Em cada tipo de iluminação, são apresentados um dos resultados sem sucesso na detecção caso existam, e outro com sucesso, isto por aumento do brilho na imagem. Estes testes foram realizados a partir das câmaras dos *smartphones* *Samsung Galaxy Core Prime*, *Huawei Y7*, *Samsung Galaxy S8* e *Samsung Galaxy Note 3*. Nestes testes apresentados foram usados biossensores com concentrações de 0, 1.5, 5 e 20 mM.

	Galaxy Core Prime		Galaxy S8		Huawei Y7	
	Jejum	Não jejum	Jejum	Não jejum	Jejum	Não jejum
Luz Natural	✓	✓	✓	✓	✓	✓
Luz Fluorescente	✓	✓	✓	✓	✓	✓
Luz Incandescente	✓	✓	✓	✗	✓	✓

Tabela C.1: Resultados da análise Colorimétrica dos biossensores vazios.

	Luz Natural		Luz Incandescente	
	Jejum	Não Jejum	Jejum	Não Jejum
1,25	✗	✓	✗	✓
5	✗	✗	✗	✗
20	✗	✓	✗	✓

Tabela C.2: Resultados da análise colorimétrica dos biossensores usando o *smartphone* *Samsung Galaxy Note 3*

C.1 Detecção do biossensor

C.1.1 Luz Solar



(a) Galaxy Prime



(b) Huawei



(c) Galaxy S8

Figura C.1: Biossensor vazio a luz natural.



(a) Galaxy Note 3

Figura C.2: Biossensor com 1.5 mM a luz natural.



(a) Galaxy Note 3

Figura C.3: Biossensor com 5 mM a luz natural.



(a) Galaxy Note 3

Figura C.4: Biossensor com 20 mM a luz natural.

C.1.1.1 Resultados da operação de brilho



(a) Galaxy Prime



(b) Huawei



(c) Galaxy S8

Figura C.5: Operações de brilho que não detetaram o sensor vazio.

(a) Galaxy Prime com $r = 4$ (b) Huawei com $r = 3$ (c) Galaxy S8 com $r = 2$

Figura C.6: Operações de brilho que detetaram o sensor vazio.

(a) Galaxy Note 3 com $r = 1$

Figura C.7: Operações de brilho que detetaram o sensor de 1.5 mM.

(a) Galaxy Note 3 com $r = 1$

Figura C.8: Operações de brilho que não detetaram o sensor de 5 mM.

APÊNDICE C. RESULTADOS DOS TESTES DE DETECÇÃO DO BIOSSENSOR E CALIBRAÇÃO DE COR



(a) Galaxy Note 3 com
 $r = 3$

Figura C.9: Operações de brilho que detetaram o sensor de 5 mM.



(a) Galaxy Note 3 com
 $r = 1$

Figura C.10: Operações de brilho que não detetaram o sensor de 20 mM.



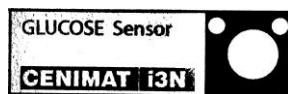
(a) Galaxy Note 3 com
 $r = 3$

Figura C.11: Operações de brilho que detetaram o sensor de 20 mM.

C.1.1.2 Resultados da operação de binarização



(a) Galaxy Prime



(b) Huawei



(c) Galaxy S8

Figura C.12: Operações de binarização que não detetaram o sensor vazio.

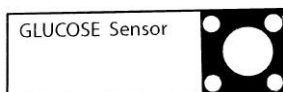
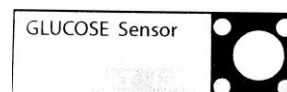
(a) Galaxy Prime com $r = 4$ (b) Huawei com $r = 3$ (c) Galaxy S8 com $r = 2$

Figura C.13: Operações de binarização que detetaram o sensor vazio.

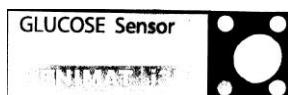
(a) Galaxy Note 3 com $r = 1$

Figura C.14: Operações de binarização que detetaram o sensor com 1.5 mM.

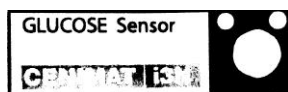
(a) Galaxy Note 3 com $r = 1$

Figura C.15: Operações de binarização que não detetaram o sensor com 5 mM.

APÊNDICE C. RESULTADOS DOS TESTES DE DETECÇÃO DO BIOSSENSOR E CALIBRAÇÃO DE COR



(a) Galaxy Note 3 com
 $r = 3$

Figura C.16: Operações de binarização que não detetaram o sensor com 5 mM.



(a) Galaxy Note 3 com
 $r = 1$

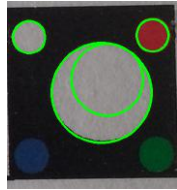
Figura C.17: Operações de binarização que não detetaram o sensor com 20 mM.



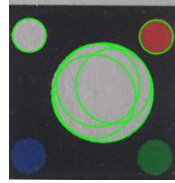
(a) Galaxy Note 3 com
 $r = 3$

Figura C.18: Operações de binarização que detetaram o sensor com 20 mM.

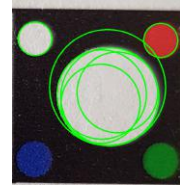
C.1.1.3 Círculos detetados



(a) Galaxy Prime

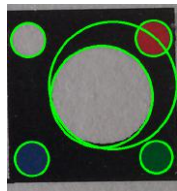


(b) Huawei

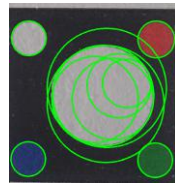


(c) Galaxy S8

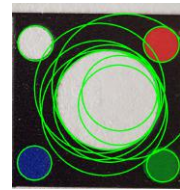
Figura C.19: Círculos totais obtidos com $r = 1$ no sensor vazio.



(a) Galaxy Prime com $r = 4$

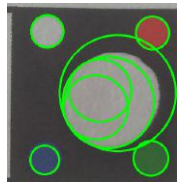


(b) Huawei com $r = 3$



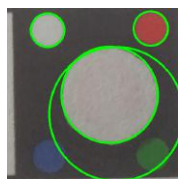
(c) Galaxy S8 com $r = 2$

Figura C.20: Círculos totais obtidos no sensor vazio.



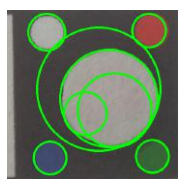
(a) Galaxy Note 3 com $r = 1$

Figura C.21: Círculos totais obtidos no sensor com 1.5 mM.



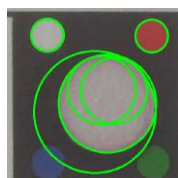
(a) Galaxy
Note 3 com
 $r = 1$

Figura C.22: Círculos totais obtidos com $r = 1$ no sensor de 5 mM.



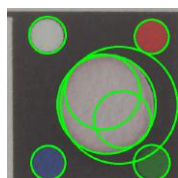
(a) Galaxy
Note 3 com
 $r = 3$

Figura C.23: Círculos totais obtidos no sensor de 5 mM.



(a) Galaxy
Note 3 com
 $r = 1$

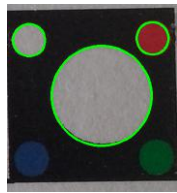
Figura C.24: Círculos totais obtidos com $r = 1$ no sensor de 20 mM.



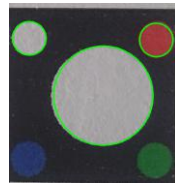
(a) Galaxy
Note 3 com
 $r = 3$

Figura C.25: Círculos totais obtidos no sensor de 20 mM.

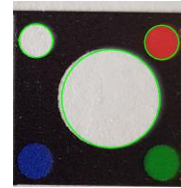
C.1.1.4 Resultados finais



(a) Galaxy Prime

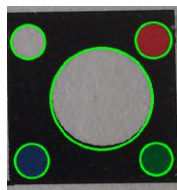


(b) Huawei

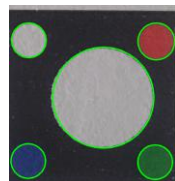


(c) Galaxy S8

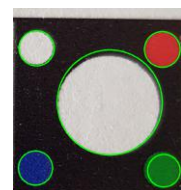
Figura C.26: Círculos finais obtidos com $r = 1$.



(a) Galaxy Prime com $r = 4$

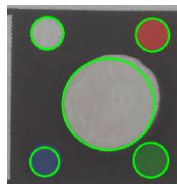


(b) Huawei com $r = 3$

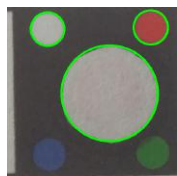


(c) Galaxy S8 com $r = 2$

Figura C.27: Círculos finais obtidos.

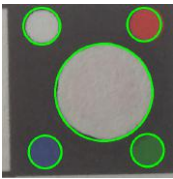


(a) Galaxy Note 3 com $r = 1$



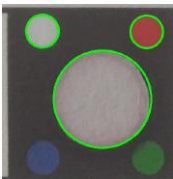
(a) Galaxy Note 3 com $r = 1$

Figura C.29: Círculos finais obtidos com $r = 1$ no sensor de 5 mM.



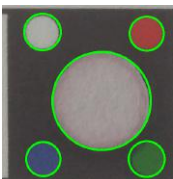
(a) Galaxy Note 3 com $r = 3$

Figura C.30: Círculos finais obtidos no sensor de 5 mM.



(a) Galaxy Note 3 com $r = 1$

Figura C.31: Círculos finais obtidos com $r = 1$ no sensor de 20 mM.



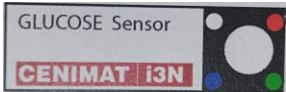
(a) Galaxy Note 3 com $r = 3$

Figura C.32: Círculos finais obtidos no sensor de 20 mM.

C.1.2 Luz Fluorescente



(a) Galaxy Prime



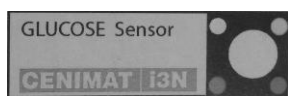
(b) Huawei



(c) Galaxy S8

Figura C.33: Biossensor vazio a luz fluorescente.

C.1.2.1 Resultados da operação de brilho



(a) Galaxy Prime



(b) Huawei



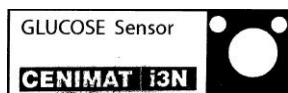
(c) Galaxy S8

Figura C.34: Operações de brilho que não detetaram o sensor.

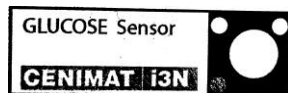
(a) Galaxy Prime com $r = 3$ (b) Huawei com $r = 2$ (c) Galaxy S8 com $r = 2$

Figura C.35: Operações de brilho que detetaram o sensor.

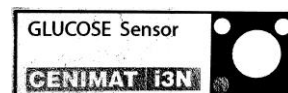
C.1.2.2 Resultados da operação de binarização



(a) Galaxy Prime



(b) Huawei



(c) Galaxy S8

Figura C.36: Operações de binarização que não detetaram o sensor.

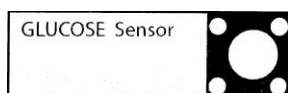
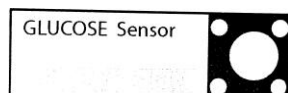
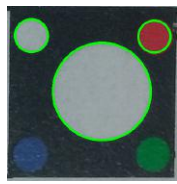
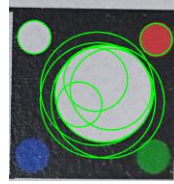
(a) Galaxy Prime com $r = 3$ (b) Huawei com $r = 2$ (c) Galaxy S8 com $r = 2$

Figura C.37: Operações de binarização que detetaram o sensor.

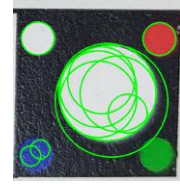
C.1.2.3 Círculos detetados



(a) Galaxy Prime

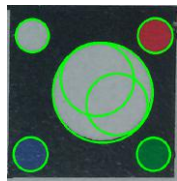


(b) Huawei

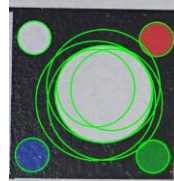


(c) Galaxy S8

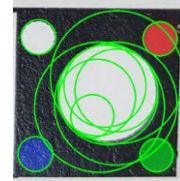
Figura C.38: Círculos totais obtidos com $r = 1$.



(a) Galaxy Prime com $r = 3$



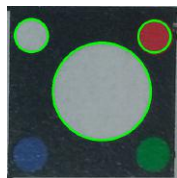
(b) Huawei com $r = 2$



(c) Galaxy S8 com $r = 2$

Figura C.39: Círculos totais obtidos.

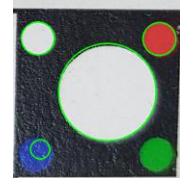
C.1.2.4 Resultados finais



(a) Galaxy Prime

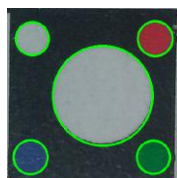


(b) Huawei

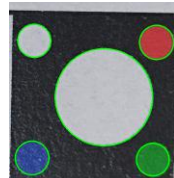


(c) Galaxy S8

Figura C.40: Círculos finais obtidos com $r = 1$.



(a) Galaxy Prime com $r = 3$



(b) Huawei com $r = 2$



(c) Galaxy S8 com $r = 2$

Figura C.41: Círculos finais obtidos.

C.1.3 Luz Incandescente



(a) Galaxy Prime



(b) Huawei



(c) Galaxy S8

Figura C.42: Biossensor vazio a luz incandescente.



(a) Galaxy Note 3

Figura C.43: Biossensor com 1.5 mM a luz natural.



(a) Galaxy Note 3

Figura C.44: Biossensor com 5 mM a luz natural.



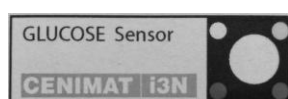
(a) Galaxy Note 3

Figura C.45: Biossensor com 20 mM a luz natural.

C.1.3.1 Resultados da operação de brilho



(a) Galaxy Prime



(b) Huawei



(c) Galaxy S8

Figura C.46: Operações de brilho com $r = 1$ que não detetou o biossensor.



(a) Galaxy Prime com $r = 3$



(b) Huawei com $r = 3$



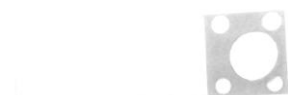
(c) Galaxy S8 com $r = 2$

Figura C.47: Operações de brilho que detetaram o sensor.



(a) Galaxy Note 3 com $r = 1$

Figura C.48: Operações de brilho que não detetaram o sensor de 1.5 mM.



(a) Galaxy Note 3 com $r = 3$

Figura C.49: Operações de brilho que detetaram o sensor de 1.5 mM.



(a) Galaxy Note 3 com
 $r = 1$

Figura C.50: Operações de brilho que não detetaram o sensor de 5 mM.



(a) Galaxy Note 3 com
 $r = 3$

Figura C.51: Operações de brilho que detetaram o sensor de 5 mM.



(a) Galaxy Note 3 com
 $r = 1$

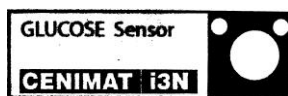
Figura C.52: Operações de brilho que não detetaram o sensor de 20 mM.



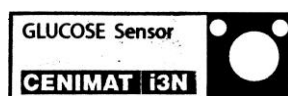
(a) Galaxy Note 3 com
 $r = 3$

Figura C.53: Operações de brilho que detetaram o sensor de 20 mM.

C.1.3.2 Resultados da operação de binarização



(a) Galaxy Prime

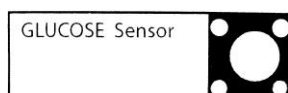


(b) Huawei



(c) Galaxy S8

Figura C.54: Operações de binarização que não detetaram o sensor.



(a) Galaxy Prime com $r = 3$



(b) Huawei com $r = 3$



(c) Galaxy S8 com $r = 2$

Figura C.55: Operações de binarização que detetaram o sensor.



(a) Galaxy Note 3 com $r = 1$

Figura C.56: Operações de binarização que não detetaram o sensor de 1.5 mM.



(a) Galaxy Note 3 com $r = 3$

Figura C.57: Operações de binarização que detetaram o sensor de 1.5 mM.



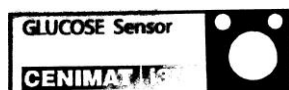
(a) Galaxy Note 3 com
 $r = 1$

Figura C.58: Operações de binarização que não detetaram o sensor de 5 mM.



(a) Galaxy Note 3 com
 $r = 3$

Figura C.59: Operações de binarização que detetaram o sensor de 5 mM.



(a) Galaxy Note 3 com
 $r = 1$

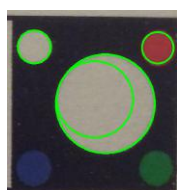
Figura C.60: Operações de binarização que não detetaram o sensor de 20 mM.



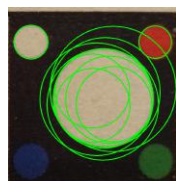
(a) Galaxy Note 3 com
 $r = 3$

Figura C.61: Operações de binarização que detetaram o sensor de 20 mM.

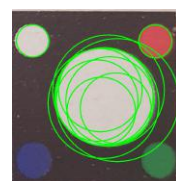
C.1.3.3 Círculos detetados



(a) Galaxy Prime

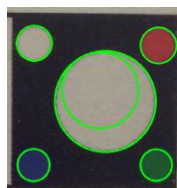


(b) Huawei

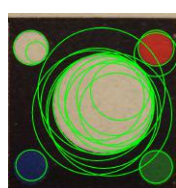


(c) Galaxy S8

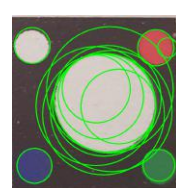
Figura C.62: Círculos totais obtidos com $r = 1$ no sensor vazio.



(a) Galaxy Prime com $r = 3$

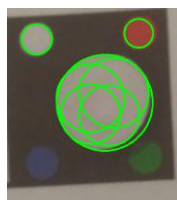


(b) Huawei com $r = 3$



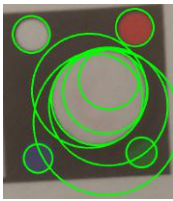
(c) Galaxy S8 com $r = 2$

Figura C.63: Círculos totais obtidos no sensor vazio.



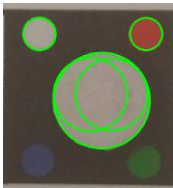
(a) Galaxy Note 3 com $r = 1$

Figura C.64: Círculos totais obtidos no sensor de 1.5 mM.



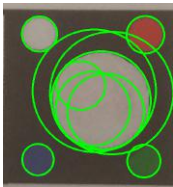
(a) Galaxy
Note 3 com
 $r = 3$

Figura C.65: Círculos totais obtidos no sensor de 1.5 mM.



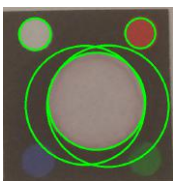
(a) Galaxy
Note 3 com
 $r = 1$

Figura C.66: Círculos totais obtidos no sensor de 5 mM.



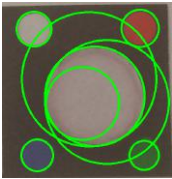
(a) Galaxy
Note 3 com
 $r = 3$

Figura C.67: Círculos totais obtidos no sensor de 5 mM.



(a) Galaxy
Note 3 com
 $r = 1$

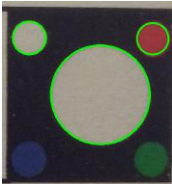
Figura C.68: Círculos totais obtidos no sensor de 20 mM.



(a) Galaxy
Note 3 com
 $r = 3$

Figura C.69: Círculos totais obtidos no sensor de 20 mM.

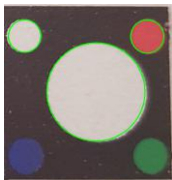
C.1.3.4 Resultados finais



(a) Galaxy
Prime

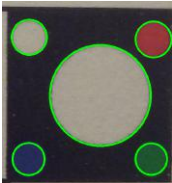


(b) Huawei

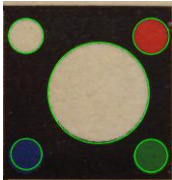


(c) Galaxy S8

Figura C.70: Círculos finais obtidos com $r = 1$.



(a) Galaxy
Prime com
 $r = 3$

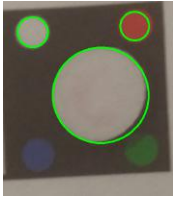


(b) Huawei
com $r = 3$



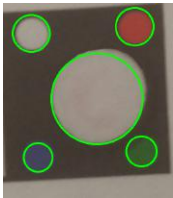
(c) Galaxy S8
com $r = 2$

Figura C.71: Círculos finais obtidos.



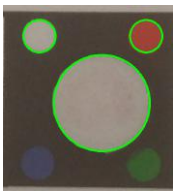
(a) Galaxy
Note 3 com
 $r = 1$

Figura C.72: Círculos finais obtidos no sensor de 1.5 mM.



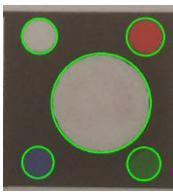
(a) Galaxy
Note 3 com
 $r = 3$

Figura C.73: Círculos finais obtidos no sensor de 1.5 mM.



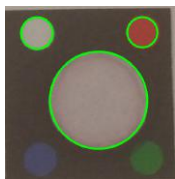
(a) Galaxy
Note 3 com
 $r = 1$

Figura C.74: Círculos finais obtidos no sensor de 5 mM.



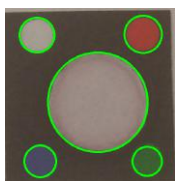
(a) Galaxy
Note 3 com
 $r = 3$

Figura C.75: Círculos finais obtidos no sensor de 5 mM.



(a) Galaxy Note 3 com $r = 1$

Figura C.76: Círculos finais obtidos no sensor de 20 mM.

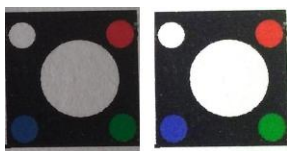


(a) Galaxy Note 3 com $r = 3$

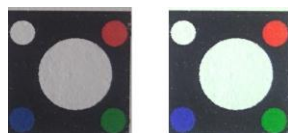
Figura C.77: Círculos finais obtidos no sensor de 20 mM.

C.2 Calibração da cor

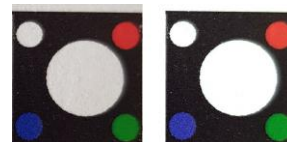
C.2.1 Luz Solar



(a) Galaxy Prime

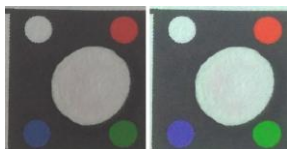


(b) Huawei



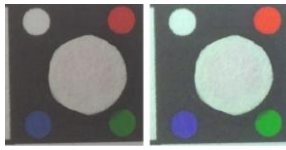
(c) Galaxy S8

Figura C.78: Comparação das cores antes e após a operação de calibração ao sensor vazio.



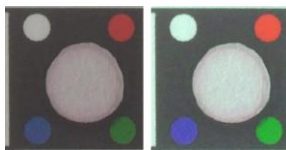
(a) Galaxy Note 3

Figura C.79: Comparação das cores antes e após a operação de calibração ao sensor com 1.5 mM.



(a) Galaxy Note 3

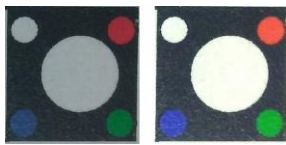
Figura C.80: Comparação das cores antes e após a operação de calibração ao sensor com 5 mM.



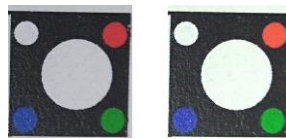
(a) Galaxy Note 3

Figura C.81: Comparação das cores antes e após a operação de calibração ao sensor com 20 mM.

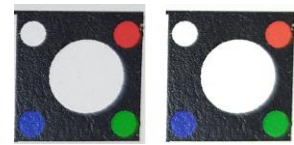
C.2.2 Luz Fluorescente



(a) Galaxy Prime



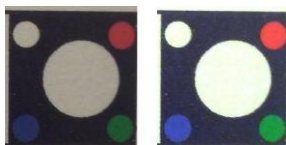
(b) Huawei



(c) Galaxy S8

Figura C.82: Comparação das cores antes e após a operação de calibração a luz fluorescente.

C.2.3 Luz Incandescente



(a) Galaxy Prime



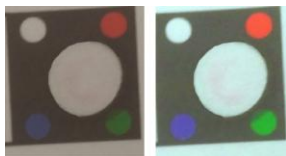
(b) Huawei



(c) Galaxy S8

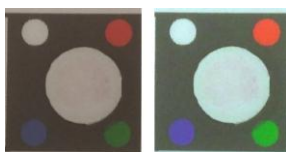
Figura C.83: Comparação das cores antes e após a operação de calibração a luz incandescente.

APÊNDICE C. RESULTADOS DOS TESTES DE DETECÇÃO DO BIOSSENSOR E CALIBRAÇÃO DE COR



(a) Galaxy Note 3

Figura C.84: Comparação das cores antes e após a operação de calibração ao sensor com 1.5 mM.



(a) Galaxy Note 3

Figura C.85: Comparação das cores antes e após a operação de calibração ao sensor com 5 mM.



(a) Galaxy Note 3

Figura C.86: Comparação das cores antes e após a operação de calibração ao sensor com 20 mM.



RESULTADOS DE *Cross-Validation*

Este apêndice apresenta os resultados da avaliação (*Precision*, *Recall* e *Fscore* com $\beta = 1$) de *Cross-Validation* para os 3 classificadores *Support Vector Machines*, *K-Nearest Neighbors* e *Random Trees*, nos diferentes formatos de cor para os conjuntos de dados de: Concentração de glucose e resultados de análise estando em jejum ou 2 horas após a refeição. Além destes formatos de cor, pretende-se também mostrar os resultados de como característica da cor, o rácio do canal R com G, sendo isso usado no trabalho desenvolvido anteriormente. De notar que para obter estes resultados, os dados foram pré-processados pelo procedimento descrito no capítulo 5.

D.1 *Cross-Validation com RGB*

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	53	57	52
K-Nearest Neighbors	70	66	65
Random Forest	57	57	56

(a) Avaliação dos modelos para classificação por concentração de glucose.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	80	80	80
K-Nearest Neighbors	88	87	87
Random Forest	85	85	85

(b) Avaliação dos modelos para classificação de resultados de Análise 2 horas após refeição.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	90	90	90
K-Nearest Neighbors	91	91	91
Random Forest	88	89	88

(c) Avaliação dos modelos para classificação de resultados de Análise em jejum.

D.2 *Cross-Validation com YUV*

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	52	46	44
K-Nearest Neighbors	57	48	47
Random Forest	52	53	52

(a) Avaliação dos modelos para classificação por concentração de glucose.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	78	77	77
K-Nearest Neighbors	81	81	81
Random Forest	82	82	82

(b) Avaliação dos modelos para classificação de resultados de Análise 2 horas após refeição.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	94	94	94
K-Nearest Neighbors	89	89	89
Random Forest	90	90	90

(c) Avaliação dos modelos para classificação de resultados de Análise em jejum.

D.3 Cross-Validation com HLS

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	53	52	47
K-Nearest Neighbors	62	59	60
Random Forest	46	48	46

(a) Avaliação dos modelos para classificação por concentração de glucose.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	80	80	80
K-Nearest Neighbors	80	80	80
Random Forest	79	77	77

(b) Avaliação dos modelos para classificação de resultados de Análise 2 horas após refeição.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	94	94	94
K-Nearest Neighbors	87	87	87
Random Forest	85	85	85

(c) Avaliação dos modelos para classificação de resultados de Análise em jejum.

D.4 Cross-Validation com HSV

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	45	54	48
K-Nearest Neighbors	55	61	56
Random Forest	52	51	50

(a) Avaliação dos modelos para classificação por concentração de glucose.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	79	78	78
K-Nearest Neighbors	84	84	83
Random Forest	81	81	81

(b) Avaliação dos modelos para classificação de resultados de Análise 2 horas após refeição.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	93	92	92
K-Nearest Neighbors	89	89	88
Random Forest	86	86	86

(c) Avaliação dos modelos para classificação de resultados de Análise em jejum.

D.5 Cross-Validation com XYZ

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	38	51	41
K-Nearest Neighbors	73	67	66
Random Forest	60	59	59

(a) Avaliação dos modelos para classificação por concentração de glucose.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	77	77	77
K-Nearest Neighbors	85	84	83
Random Forest	88	87	87

(b) Avaliação dos modelos para classificação de resultados de Análise 2 horas após refeição.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	91	91	91
K-Nearest Neighbors	89	87	87
Random Forest	92	92	92

(c) Avaliação dos modelos para classificação de resultados de Análise em jejum.

D.6 Cross-Validation com o rácio de R com G

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	36	38	31
K-Nearest Neighbors	43	42	41
Random Forest	48	46	45

(a) Avaliação dos modelos para classificação por concentração de glucose.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	76	75	70
K-Nearest Neighbors	77	77	77
Random Forest	78	78	78

(b) Avaliação dos modelos para classificação de resultados de Análise 2 horas após refeição.

Classificador	Precision (%)	Recall (%)	Fscore (%)
Support Vector Machines	78	80	79
K-Nearest Neighbors	80	81	81
Random Forest	86	86	86

(c) Avaliação dos modelos para classificação de resultados de Análise em jejum.

REGIÕES DE DECISÃO DOS MODELOS DESENVOLVIDOS

Neste anexo encontram-se as diferentes regiões de decisão obtidas aquando dos testes de *Cross-Validation* de modo a determinar os melhores modelos a serem usados nesta dissertação, para os três classificadores *Support Vector Machines*, *K-Nearest Neighbors* e *Random Trees*. Na tabela E.1 pode-se verificar o significado das etiquetas apresentadas em cada um dos gráficos.

Etiqueta	Concentração
0	0
1	1.25
2	1.5
3	2
4	5
5	7
6	10
7	12.5
8	15
9	20

(a) Concentrações de glucose.

Etiqueta	Resultado da análise
0	Sensor Vazio
1	Hipoglicemia
2	Normal ou Pré-Diabetes
3	Diabetes

(b) Resultados de análise.

Tabela E.1: Significado das etiquetas relativas aos gráficos das regiões de decisão.

E.1 Modelo de cor RGB

E.1.1 Concentrações de glucose

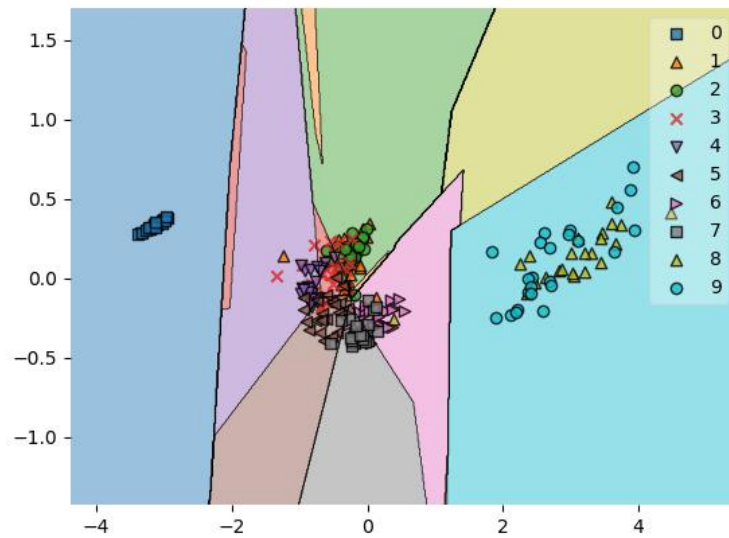


Figura E.1: *Support Vector Machines*

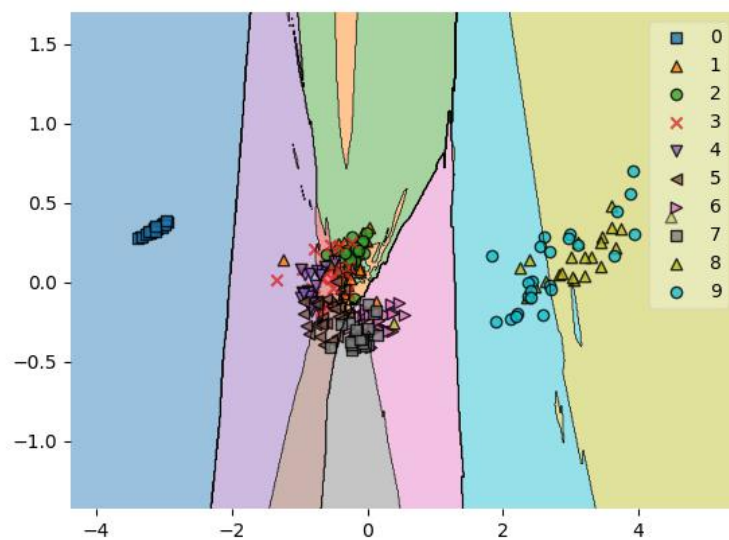
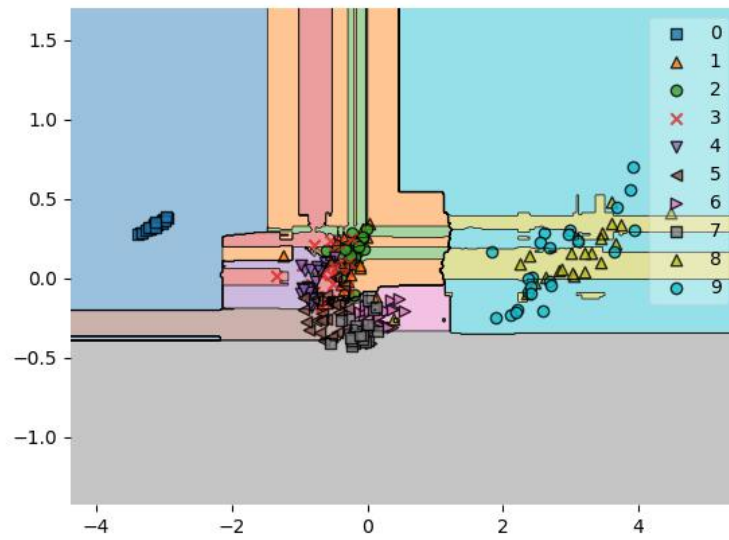
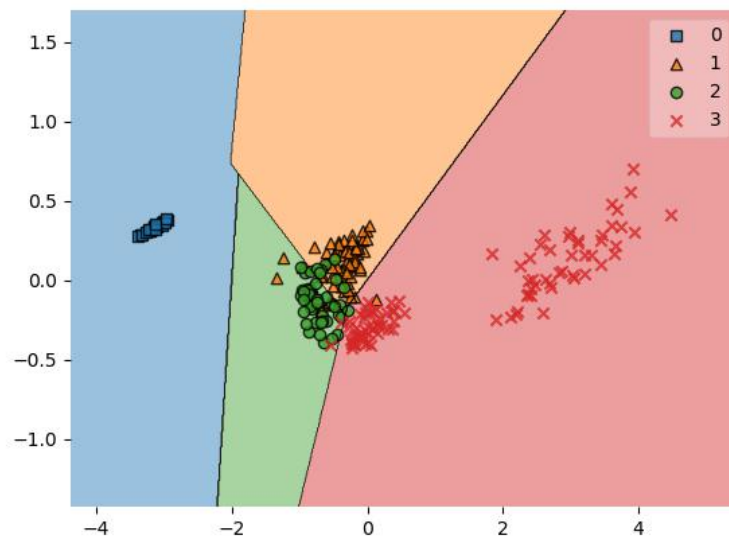


Figura E.2: *K-Nearest Neighbors*

Figura E.3: *Random Trees*

E.1.2 Estado de Jejum

Figura E.4: *Support Vector Machines*

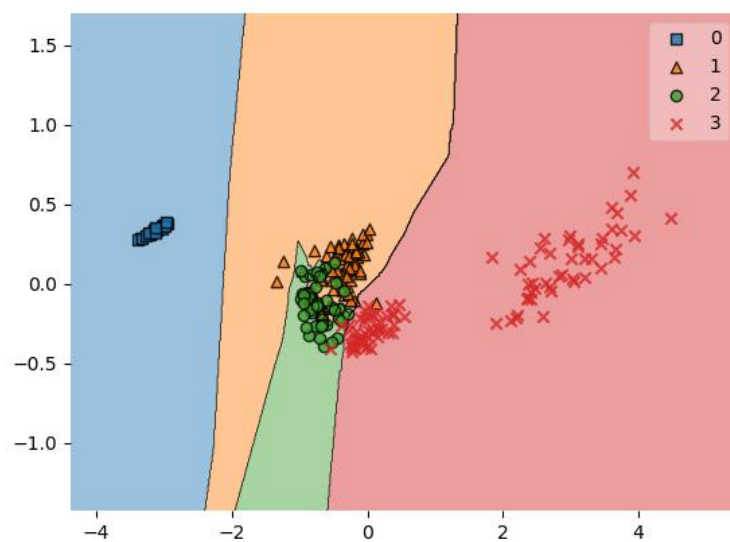


Figura E.5: *K-Nearest Neighbors*

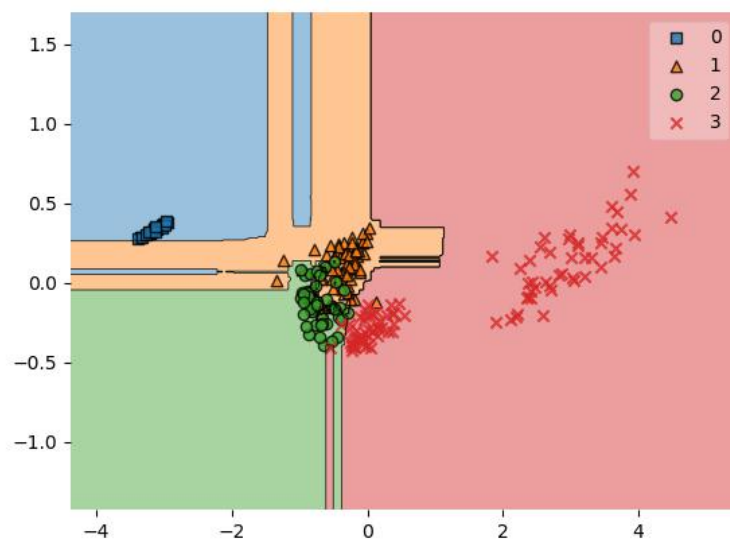
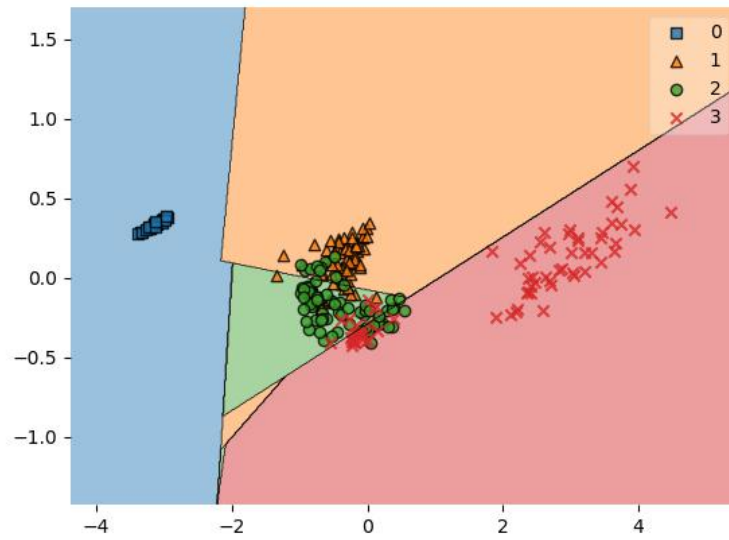
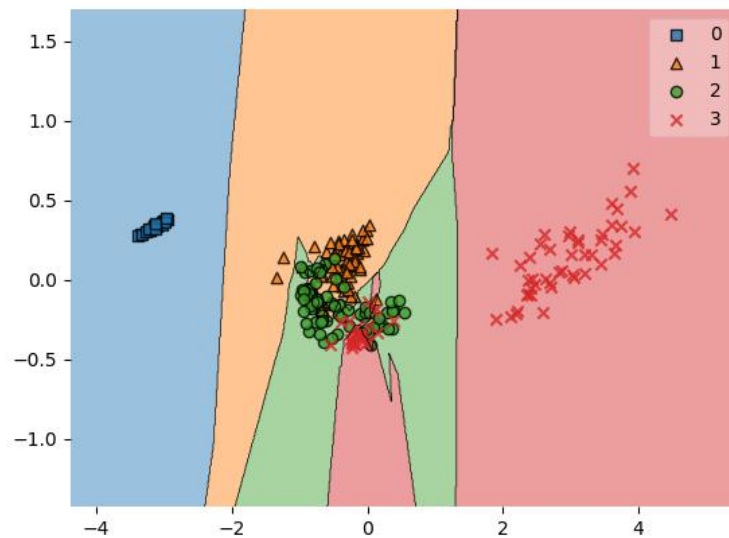


Figura E.6: *Random Trees*

E.1.3 Duas horas após a refeição

Figura E.7: *Support Vector Machines*Figura E.8: *K-Nearest Neighbors*

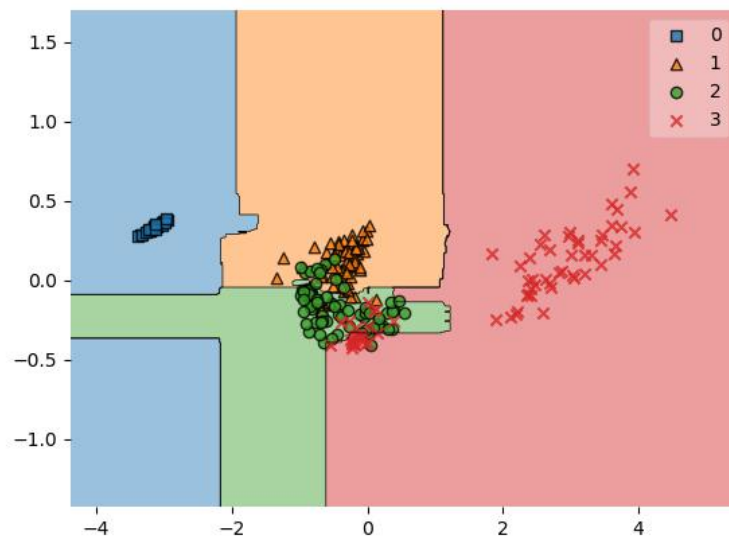


Figura E.9: *Random Trees*

E.2 Modelo de cor YUV

E.2.1 Concentrações de glucose

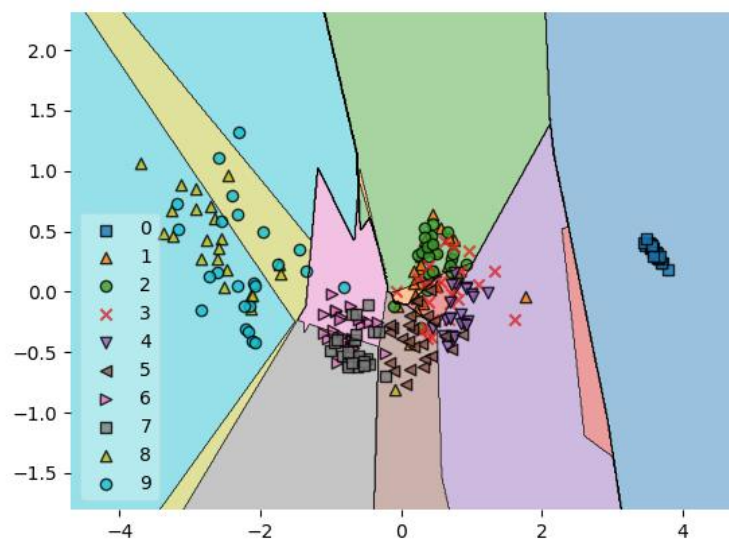


Figura E.10: *Support Vector Machines*

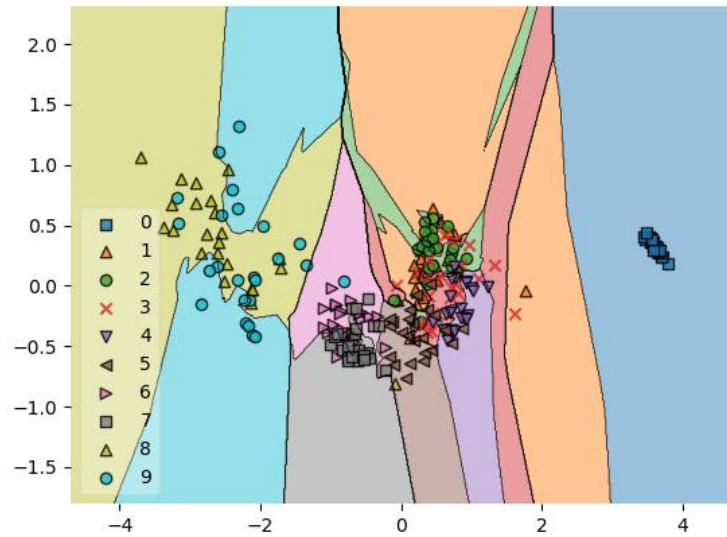


Figura E.11: *K-Nearest Neighbors*

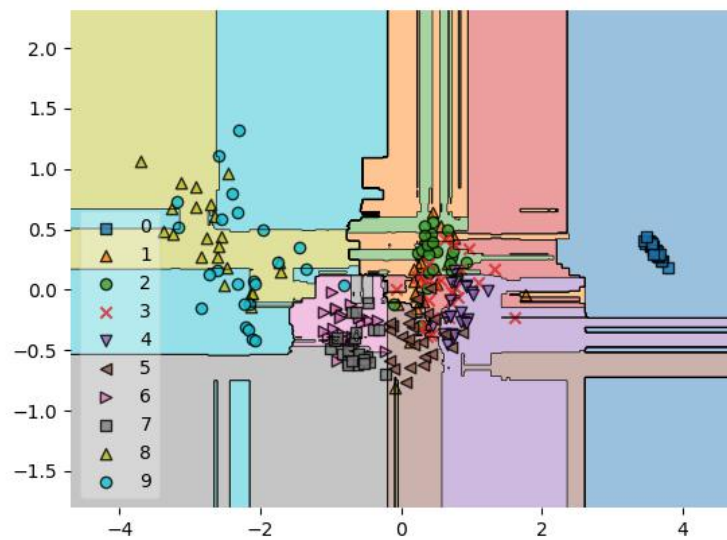


Figura E.12: *Random Trees*

E.2.2 Estado de Jejum

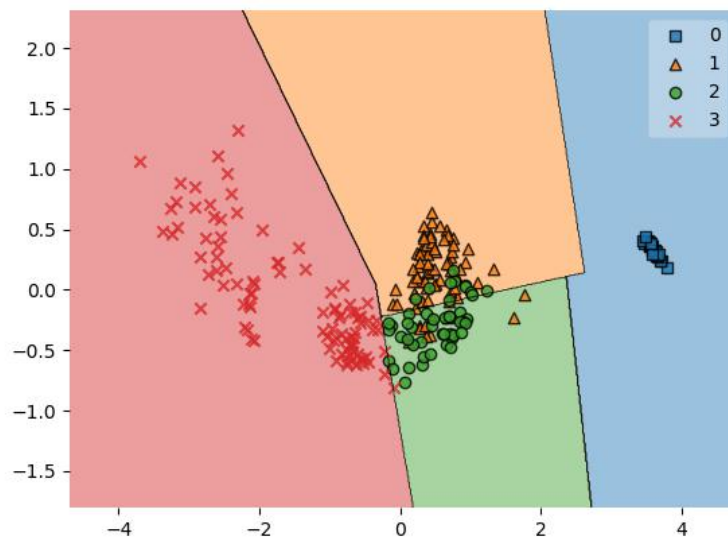


Figura E.13: *Support Vector Machines*

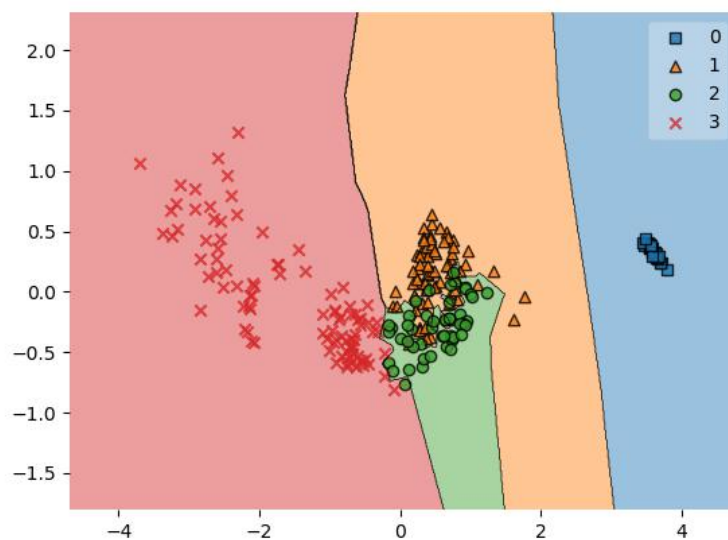
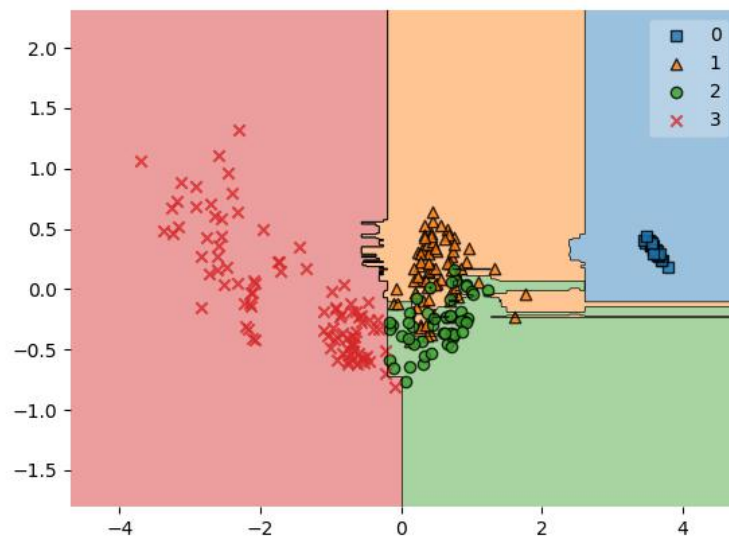
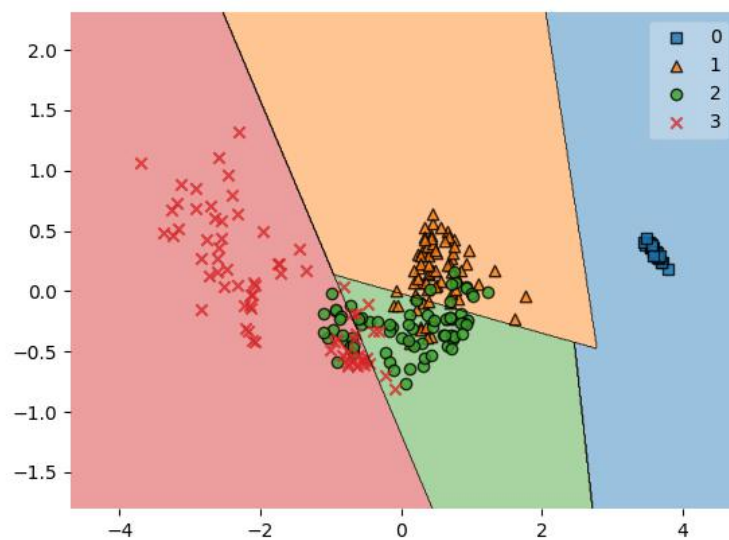


Figura E.14: *K-Nearest Neighbors*

Figura E.15: *Random Trees*

E.2.3 Duas horas após a refeição

Figura E.16: *Support Vector Machines*

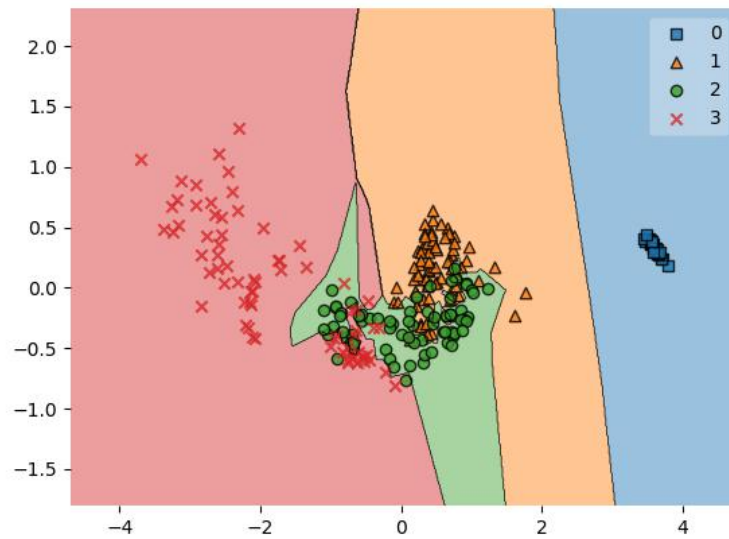


Figura E.17: *K-Nearest Neighbors*

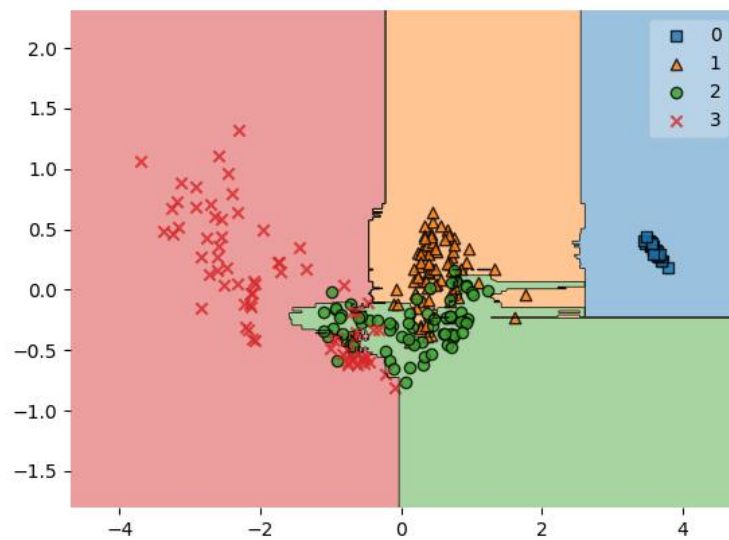


Figura E.18: *Random Trees*

E.3 Modelo de cor HSV

E.3.1 Concentrações de glucose

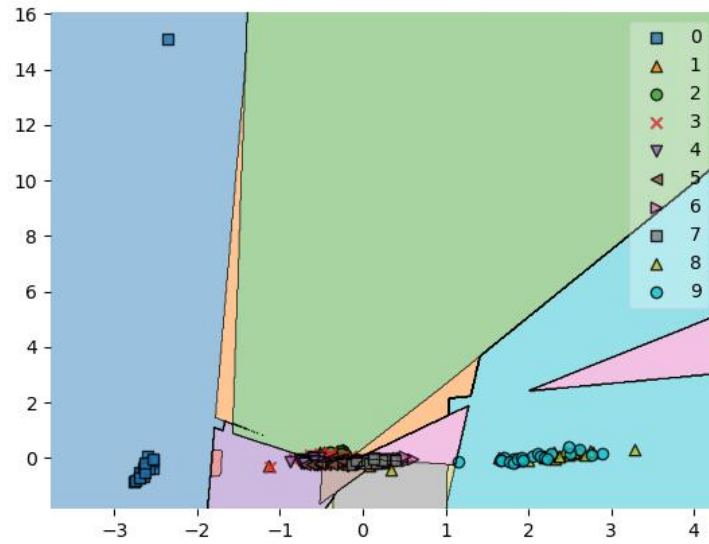


Figura E.19: *Support Vector Machines*

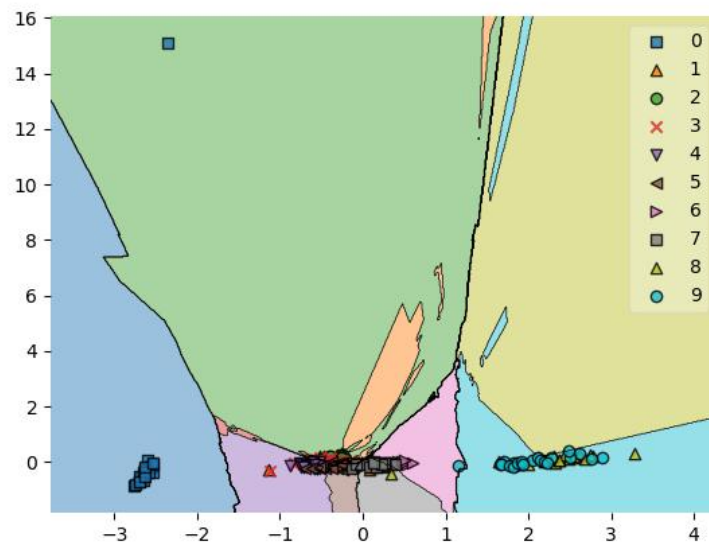


Figura E.20: *K-Nearest Neighbors*

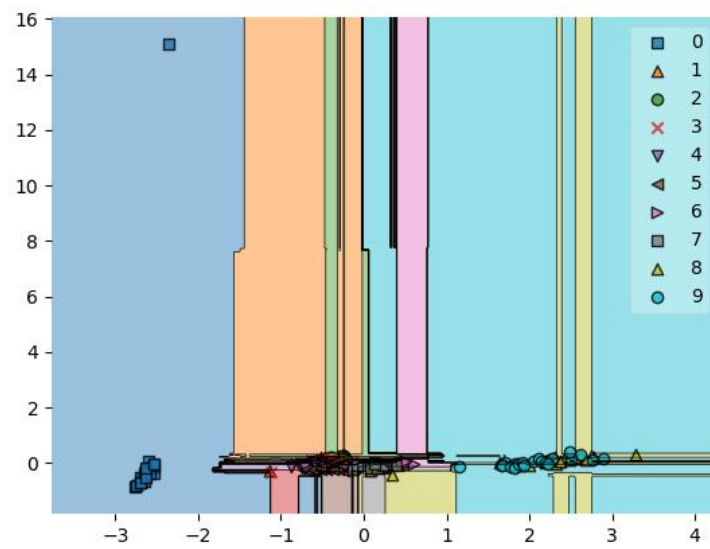


Figura E.21: *Random Trees*

E.3.2 Estado de Jejum

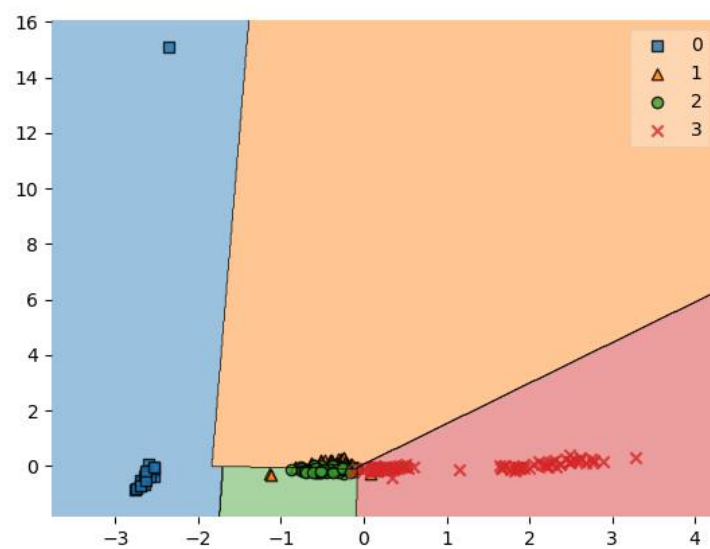


Figura E.22: *Support Vector Machines*

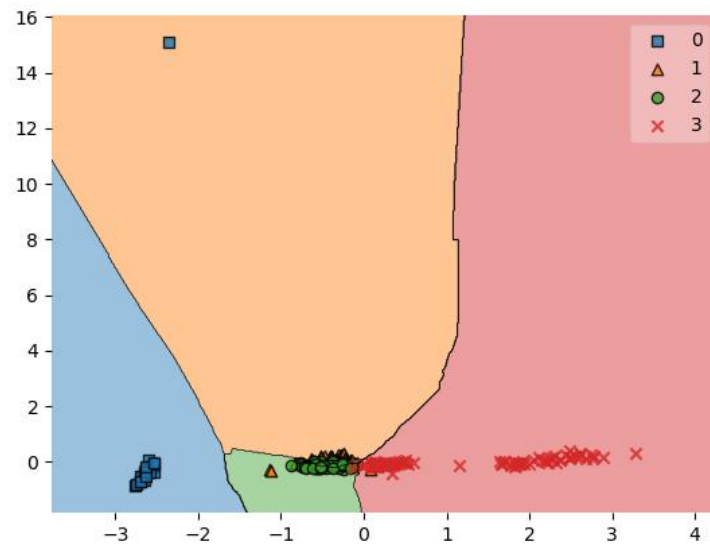


Figura E.23: *K-Nearest Neighbors*

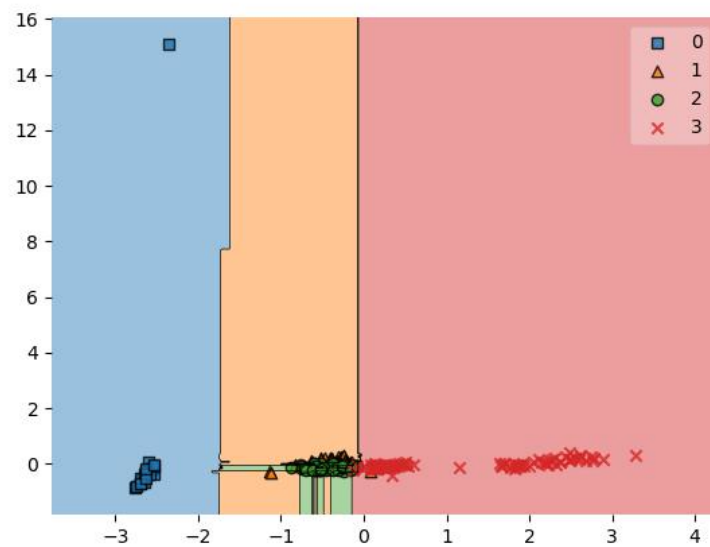


Figura E.24: *Random Trees*

E.3.3 Duas horas após a refeição

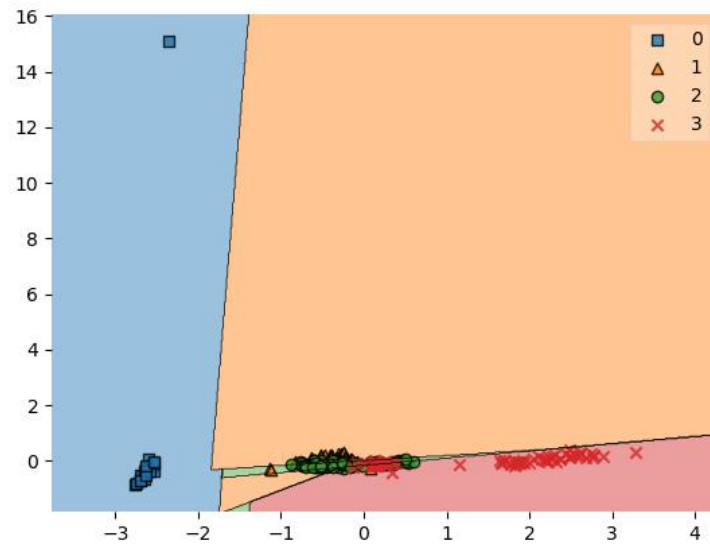


Figura E.25: *Support Vector Machines*

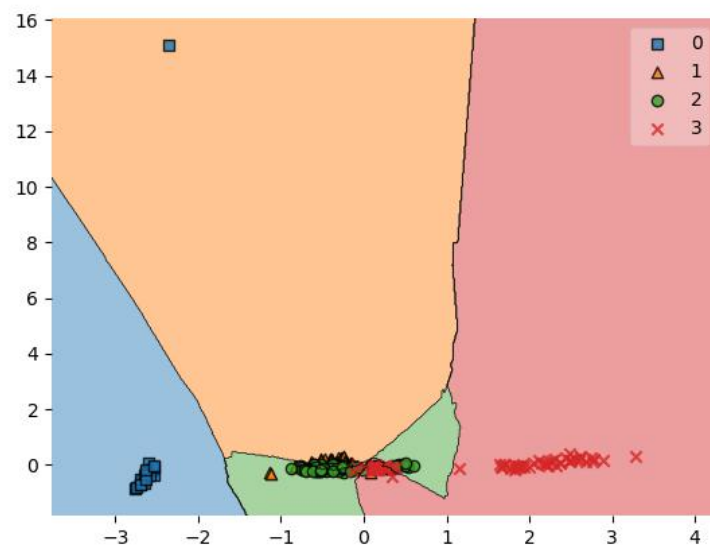
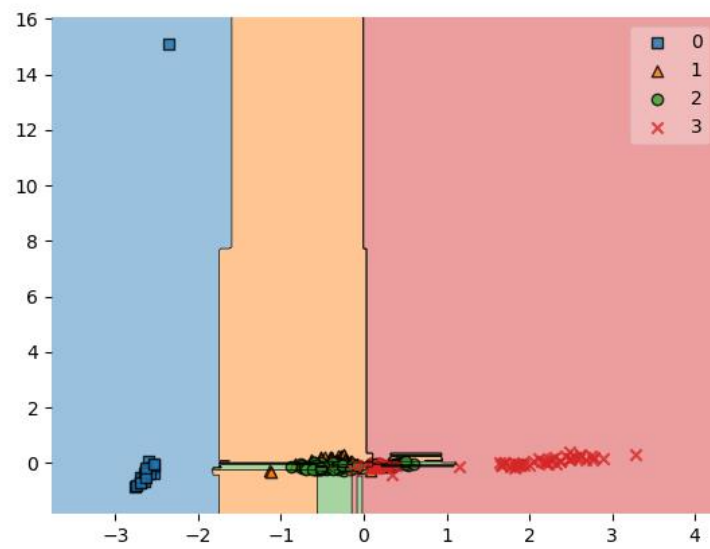
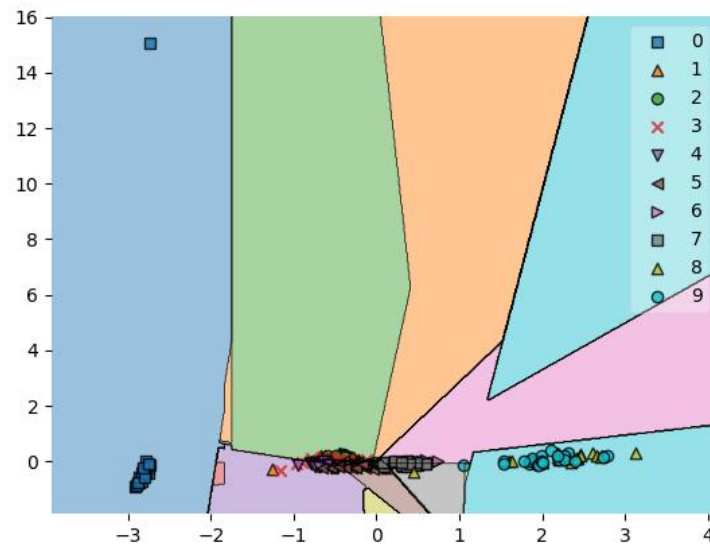


Figura E.26: *K-Nearest Neighbors*

Figura E.27: *Random Trees*

E.4 Modelo de cor HLS

E.4.1 Concentrações de glucose

Figura E.28: *Support Vector Machines*

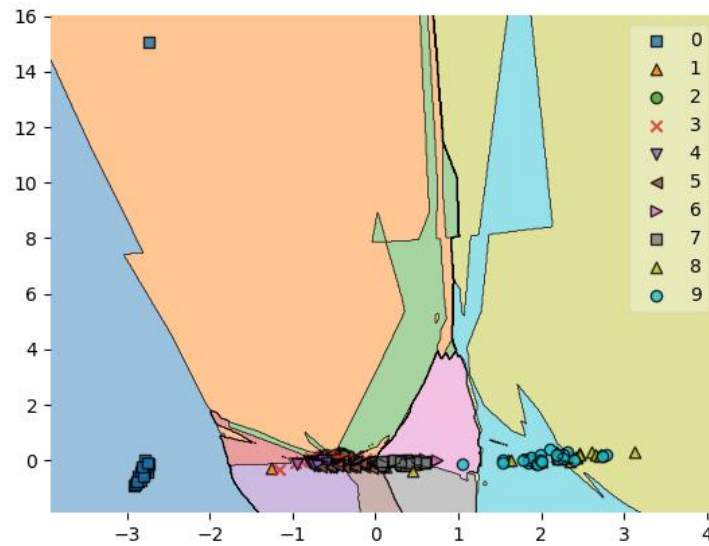


Figura E.29: *K-Nearest Neighbors*

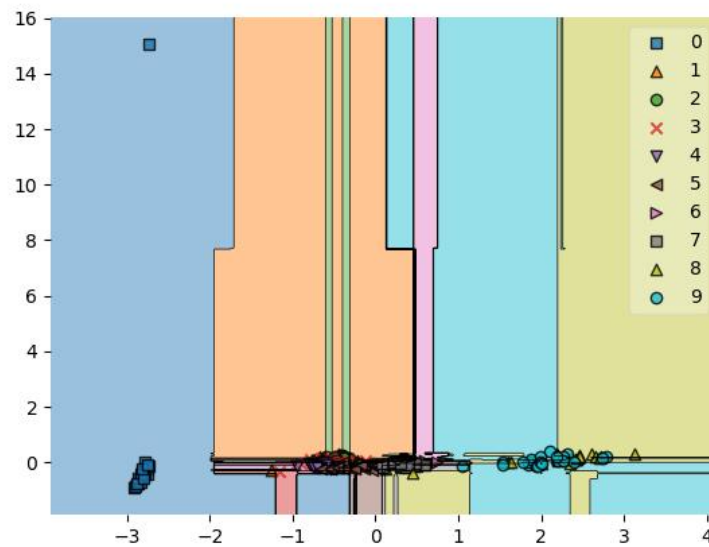
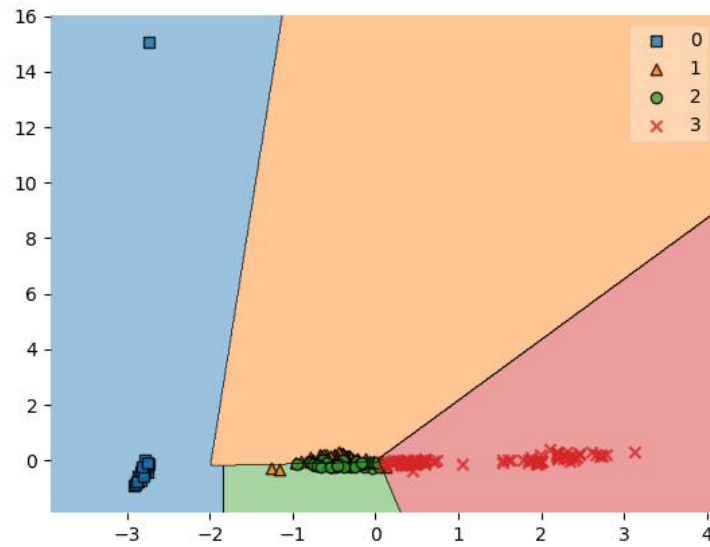
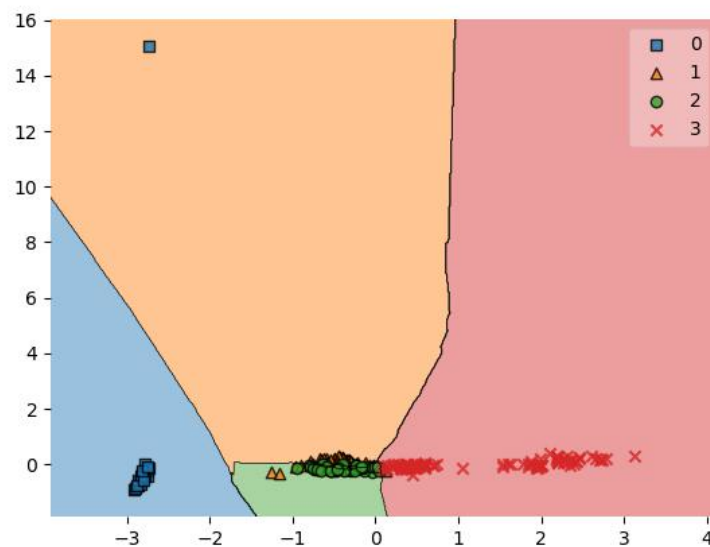


Figura E.30: *Random Trees*

E.4.2 Estado de Jejum

Figura E.31: *Support Vector Machines*Figura E.32: *K-Nearest Neighbors*

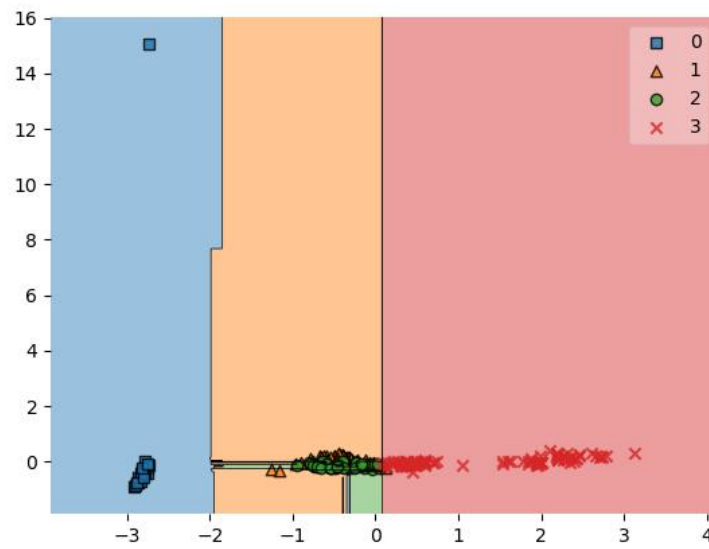


Figura E.33: *Random Trees*

E.4.3 Duas horas após a refeição

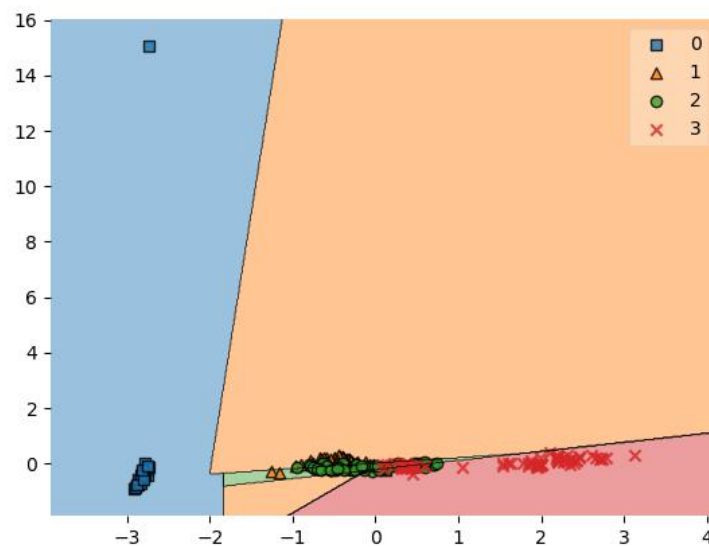


Figura E.34: *Support Vector Machines*

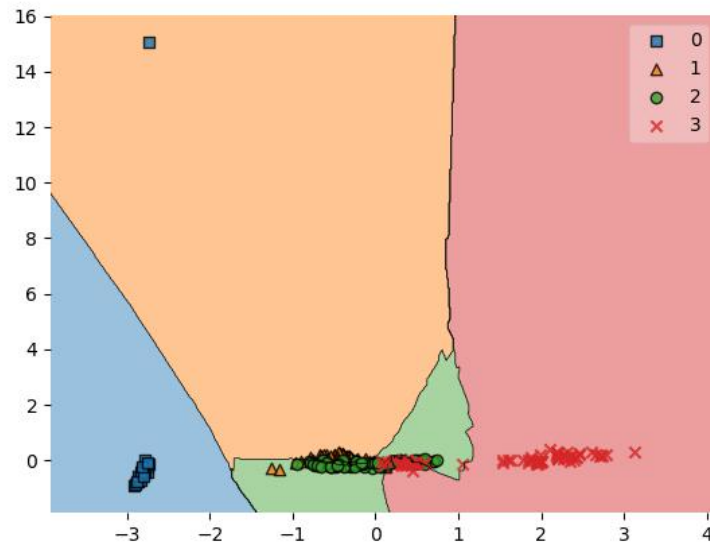


Figura E.35: *K-Nearest Neighbors*

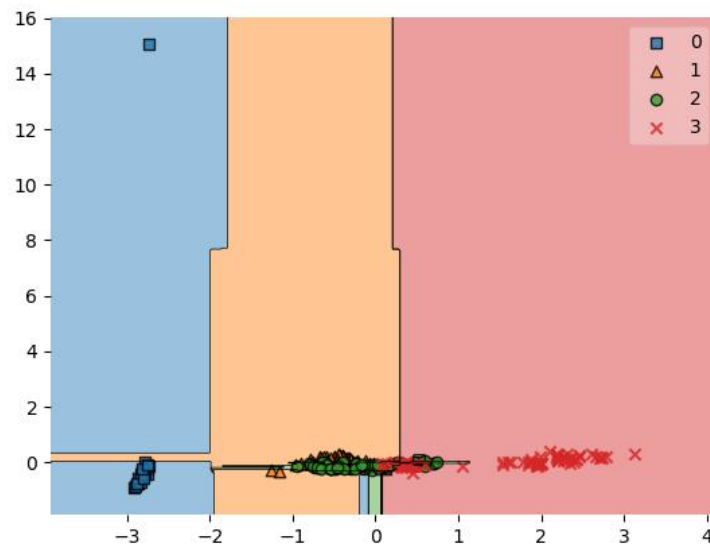


Figura E.36: *Random Trees*

E.5 Modelo de cor XYZ

E.5.1 Concentrações de glucose

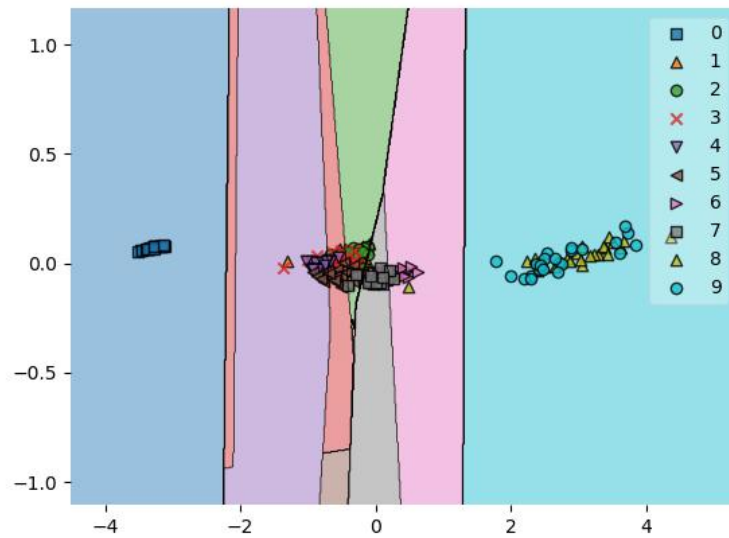


Figura E.37: *Support Vector Machines*

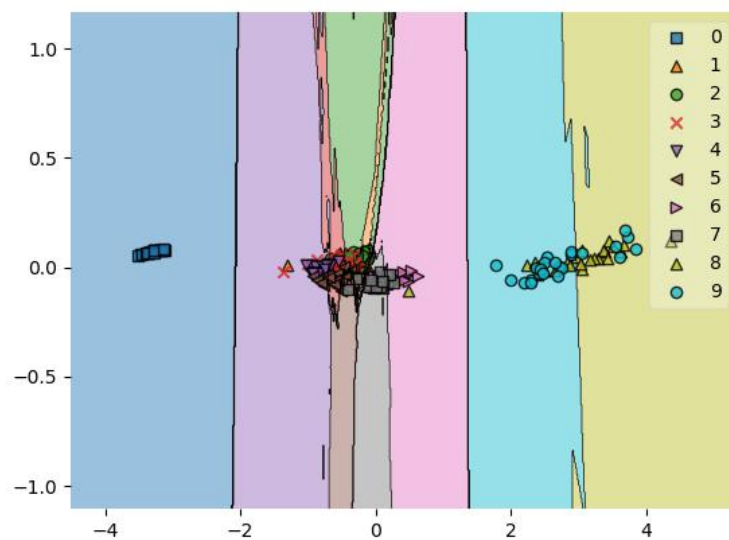
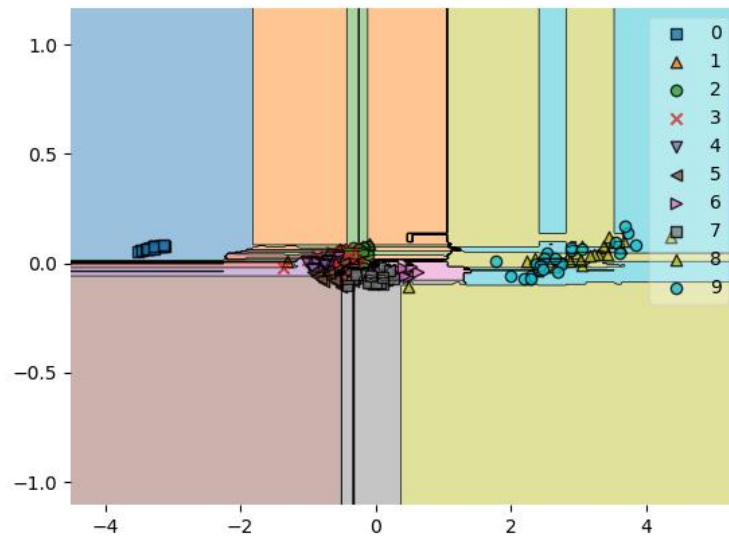
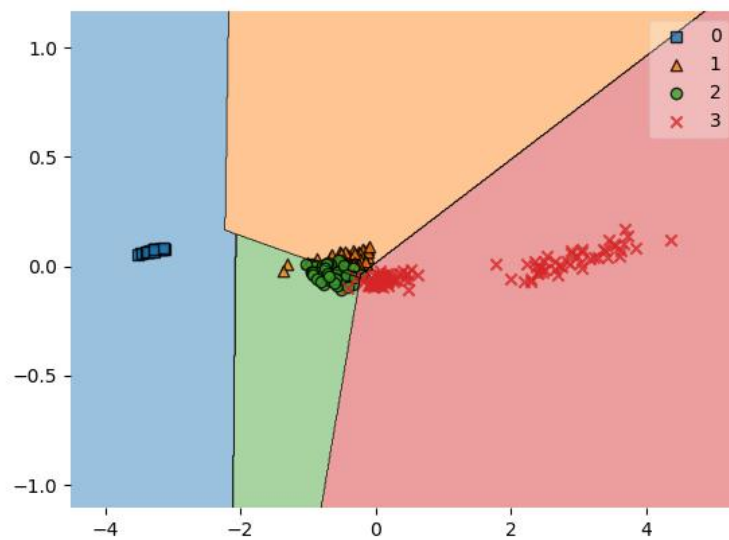


Figura E.38: *K-Nearest Neighbors*

Figura E.39: *Random Trees*

E.5.2 Estado de Jejum

Figura E.40: *Support Vector Machines*

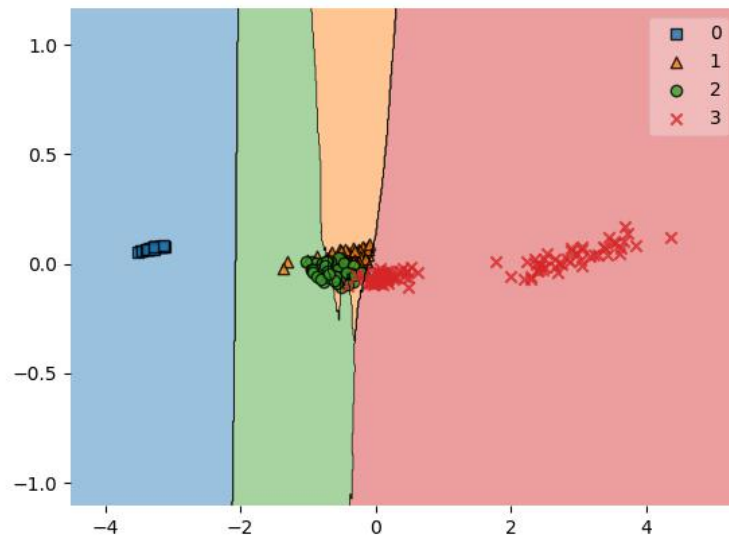


Figura E.41: *K-Nearest Neighbors*

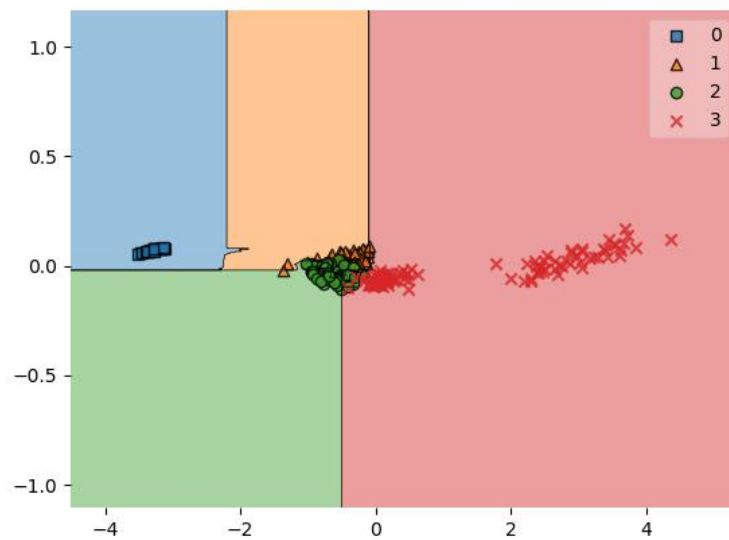


Figura E.42: *Random Trees*

E.5.3 Duas horas após a refeição

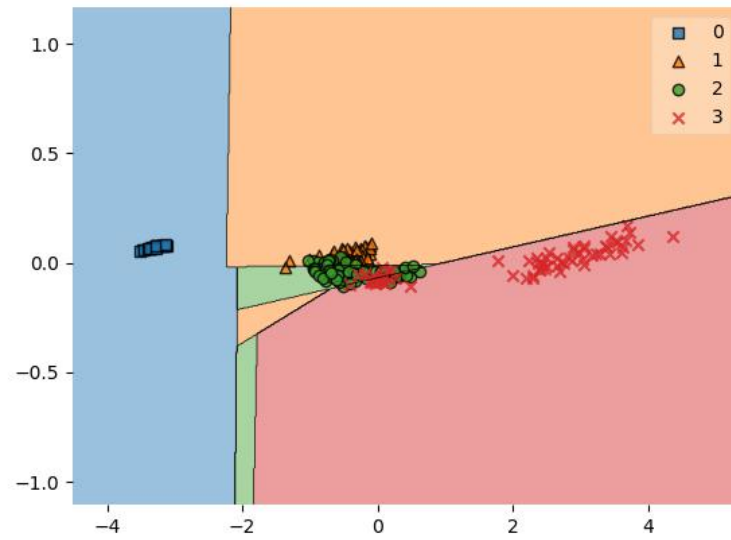


Figura E.43: *Support Vector Machines*

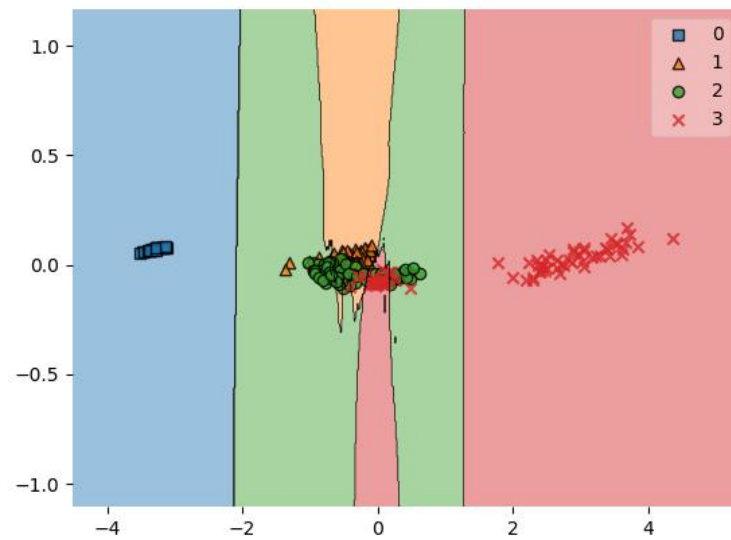


Figura E.44: *K-Nearest Neighbors*

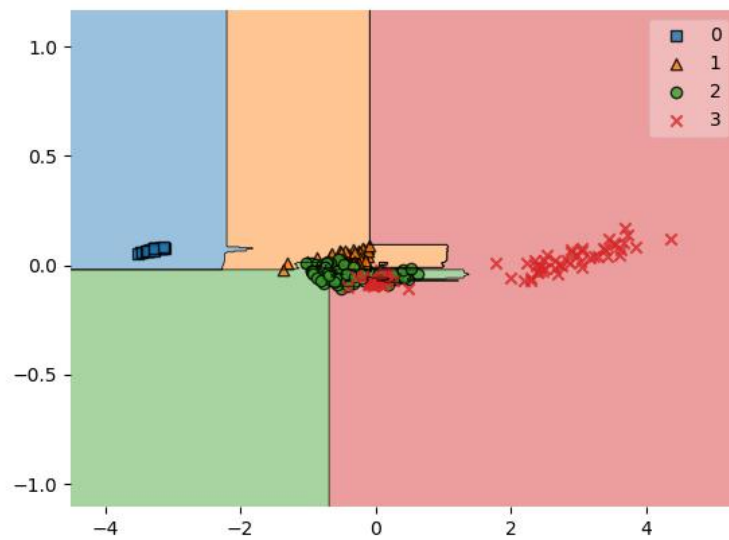


Figura E.45: *Random Trees*

E.6 Modelo a partir do rácio de R com G

E.6.1 Concentrações de glucose

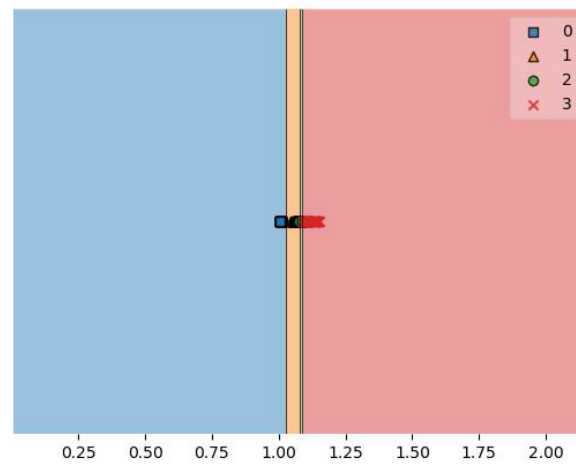


Figura E.46: *Support Vector Machines*

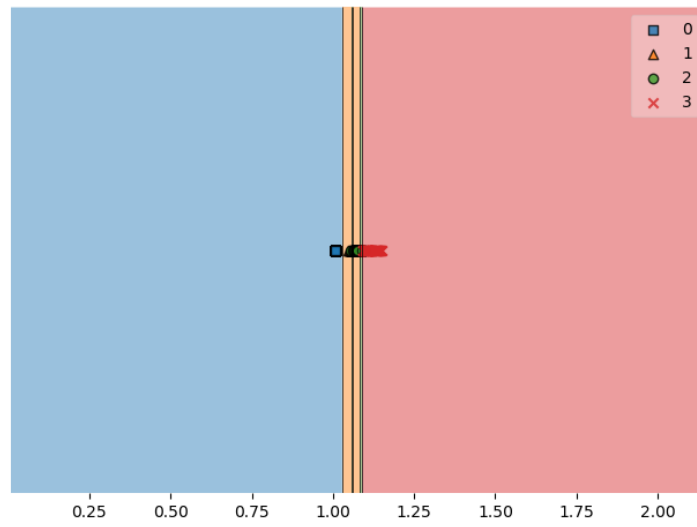


Figura E.47: *K-Nearest Neighbors*

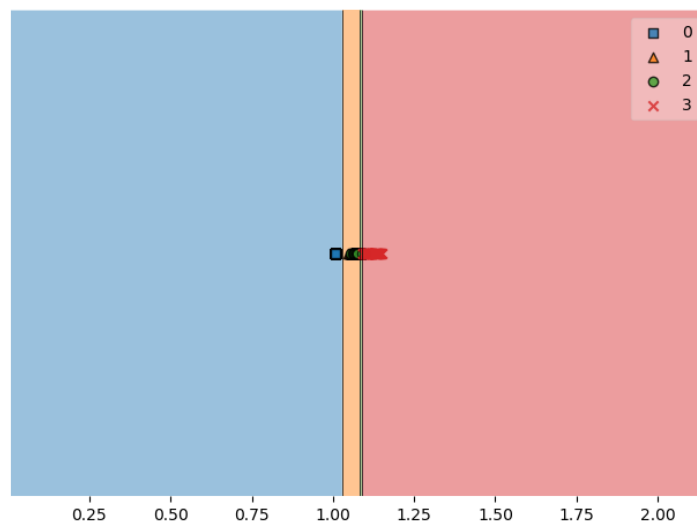


Figura E.48: *Random Trees*

E.6.2 Estado de Jejum

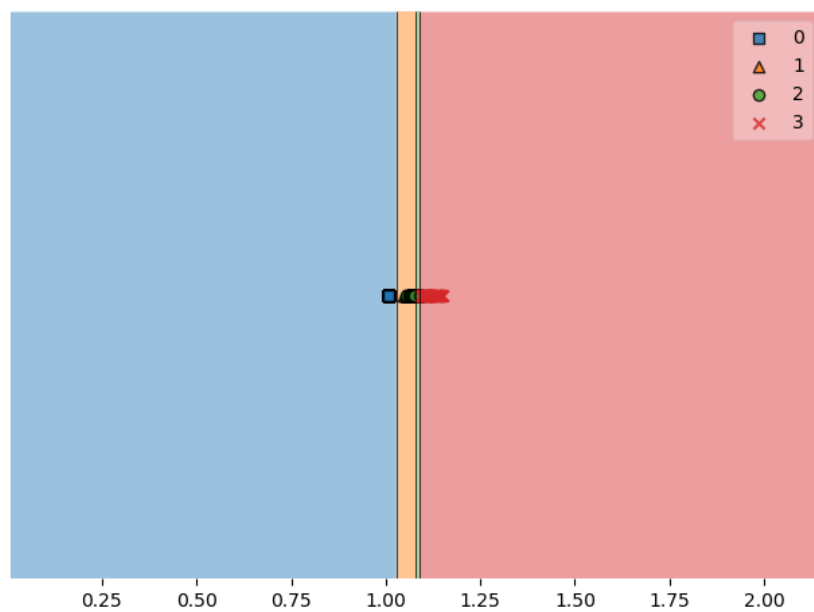


Figura E.49: *Support Vector Machines*

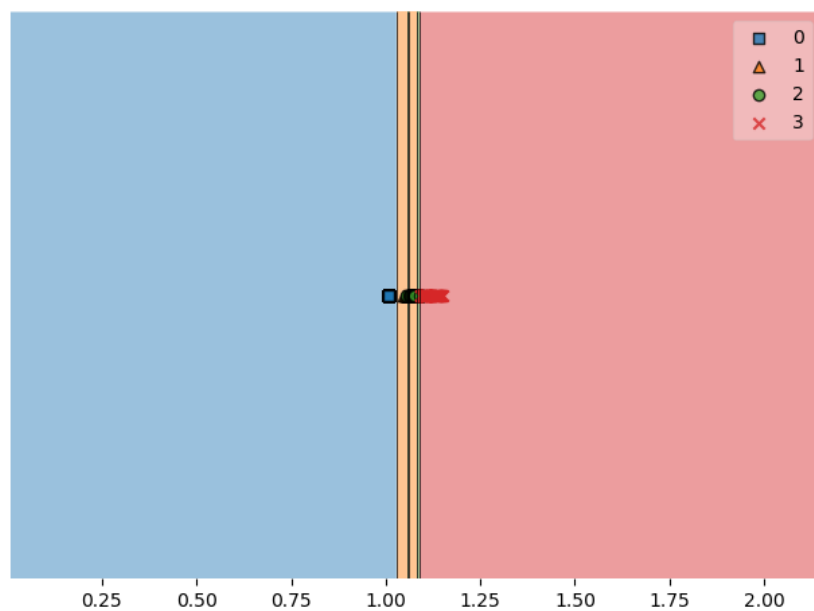


Figura E.50: *K-Nearest Neighbors*

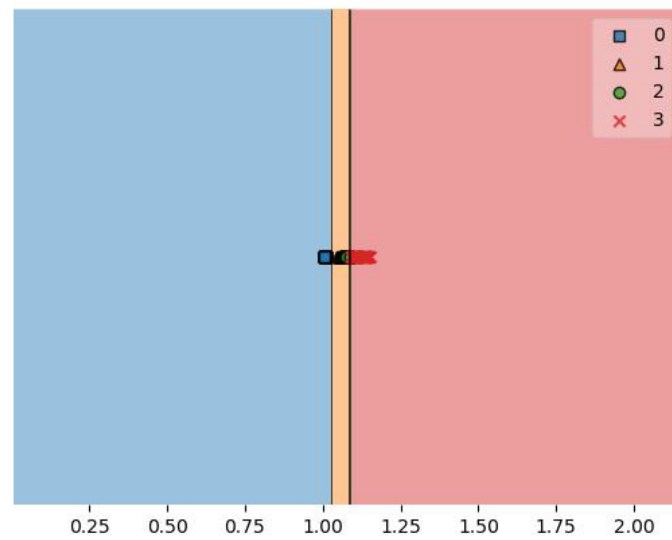


Figura E.51: *Random Trees*

E.6.3 Duas horas após a refeição

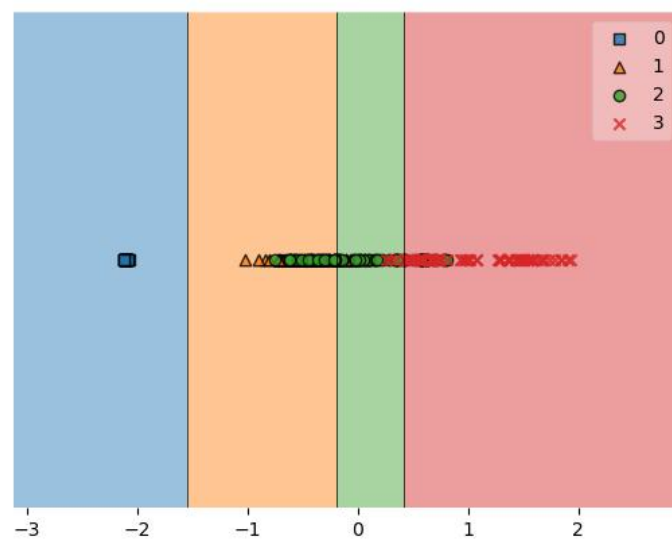


Figura E.52: *Support Vector Machines*

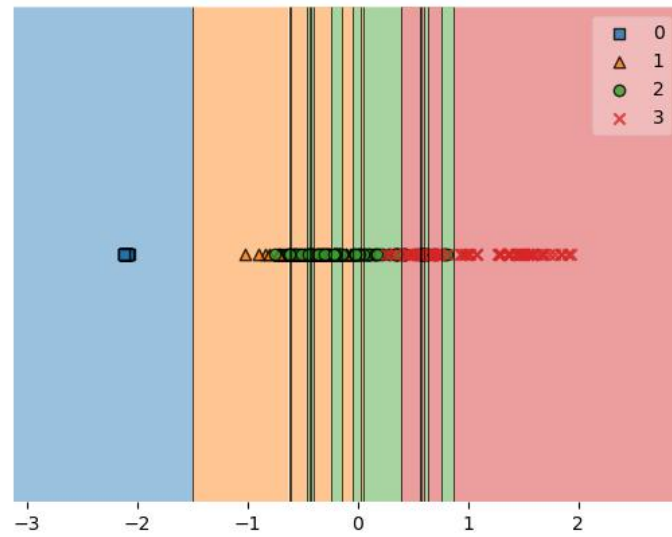


Figura E.53: *K-Nearest Neighbors*

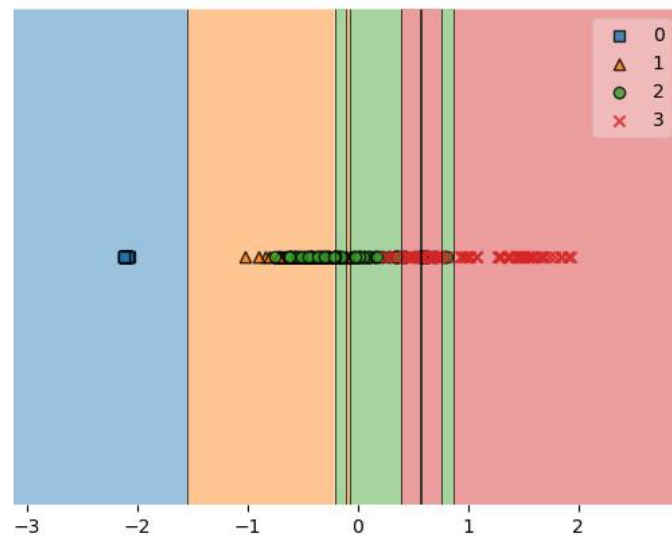


Figura E.54: *Random Trees*



QUESTIONÁRIO DE USABILIDADE

Neste apêndice encontra-se o questionário de usabilidade usado para realizar a avaliação no contexto de utilizadores, em conjunto com a percentagem de respostas que se obteve em cada pergunta. De notar que na escala usada para as perguntas relacionadas com a interface, a escala vai de ordem crescente de concordância: Discordo completamente, Discordo ligeiramente, Concordo, Concordo ligeiramente e Concordo completamente.

F.1 Qual a sua idade?

- 18-24 anos
- 25-34 anos
- 35-44 anos
- 45-54 anos
- +55 anos

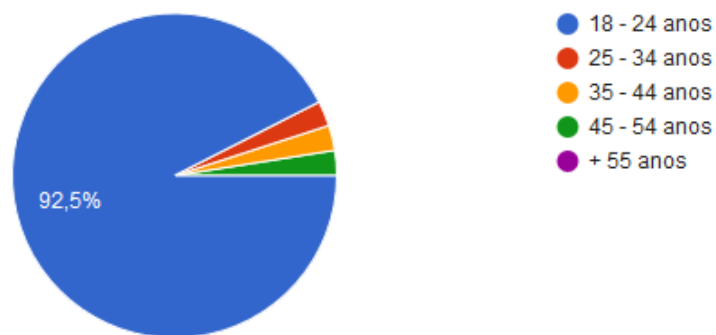


Figura F.1

F.2 É Diabético/a?

- Sim
- Não
- Não sei

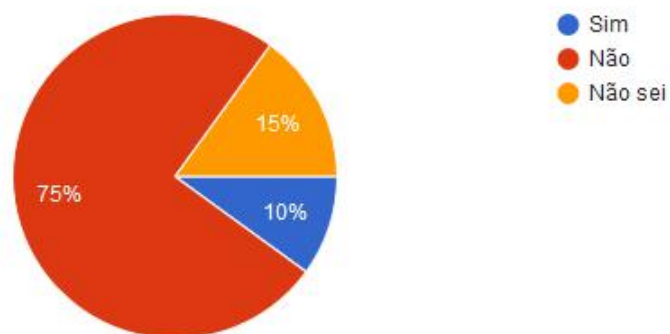


Figura F.2

F.3 Acho que a deteção do biossensor é fácil

- Concordo completamente
- Concordo ligeiramente
- Concordo
- Discordo ligeiramente

F.4. INICIALMENTE TIVE DIFICULDADE EM DESCOBRIR COMO DETETAR O SENSOR

- Discordo completamente

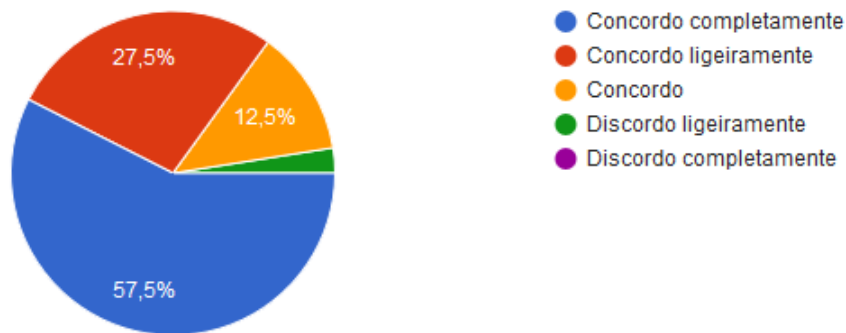


Figura F.3

F.4 Inicialmente tive dificuldade em descobrir como detetar o sensor

- Concordo completamente
- Concordo ligeiramente
- Concordo
- Discordo ligeiramente
- Discordo completamente

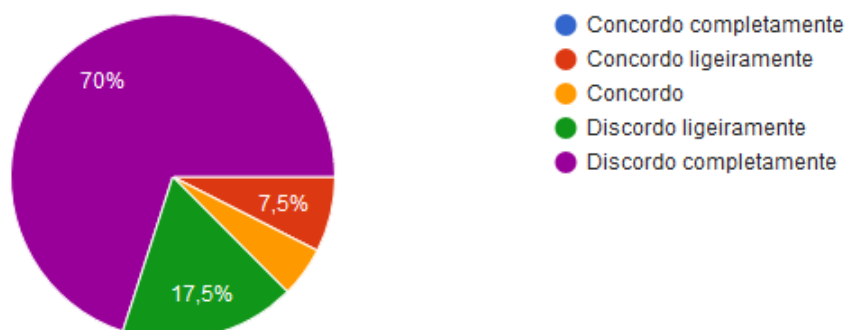


Figura F.4

F.5 A informação da análise é útil

- Concordo completamente
- Concordo ligeiramente
- Concordo
- Discordo ligeiramente
- Discordo completamente

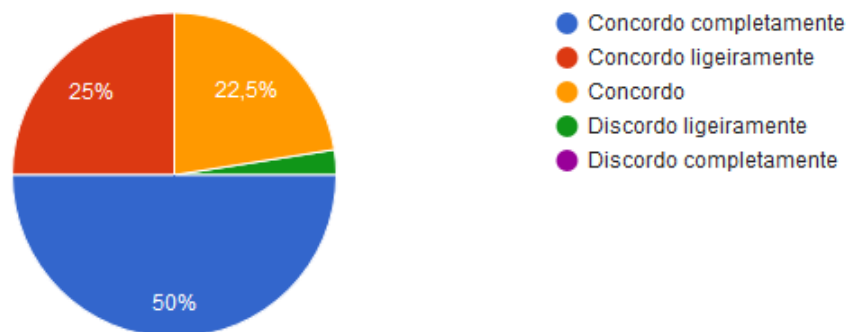


Figura F.5

F.6 A aplicação é mais fácil de se usar que uma máquina de diabetes normal

- Concordo completamente
- Concordo ligeiramente
- Concordo
- Discordo ligeiramente
- Discordo completamente

F.7. OS RESULTADOS DA ANÁLISE DA APLICAÇÃO SÃO MAIS ÚTEIS QUE OS DE UMA MÁQUINA NORMAL

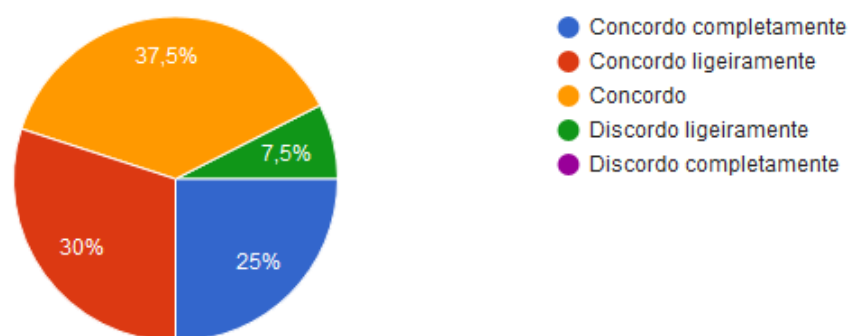


Figura F.6

F.7 Os resultados da análise da aplicação são mais úteis que os de uma máquina normal

- Concordo completamente
- Concordo ligeiramente
- Concordo
- Discordo ligeiramente
- Discordo completamente

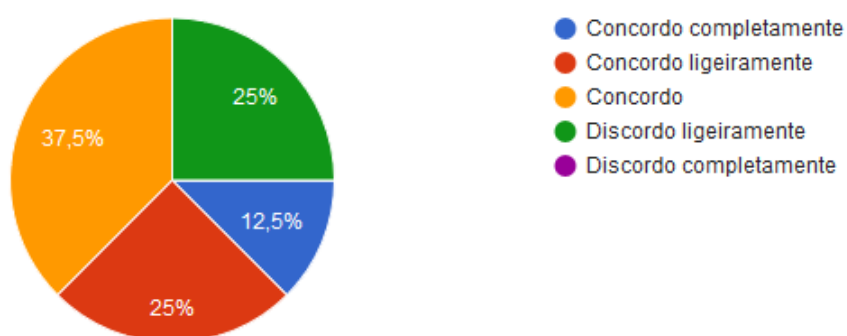


Figura F.7

F.8 Considero que a aplicação facilita a análise aos diabetes

- Concordo completamente
- Concordo ligeiramente
- Concordo
- Discordo ligeiramente
- Discordo completamente

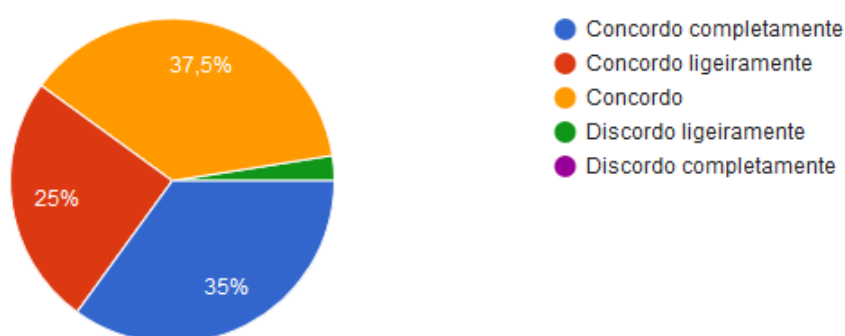


Figura F.8